PATENT ABSTRACTS OF JAPAN

(11)Publication number : **07-038555**
(43)Date of publication of application : **07.02.1995**

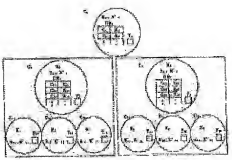
(51)Int.Cl. H04L 9/00
G09C 1/00

(21)Application number : **05-155579**
(22)Date of filing : **25.06.1993**

(71)Applicant : **SONY CORP**
(72)Inventor : **FUJINAMI YORIHISA**

(54) PUBLIC KEY CRYPTOGRAPHIC PROCESSING SYSTEM AND METHOD THEREFOR

(57)Abstract:
PURPOSE: To provide a public key cryptographic processing system keeping security and genuineness even when a computer is moved in the public key cryptographic processing system to communicate a communication sentence by performing cryptographic processing while using a public key.
CONSTITUTION: In the public key cryptographic processing system to perform a cryptographic-processed communication while using a public key K0 at plural computers C0, C1 and C2 as communication media, the key to be used for cryptography in the computer is managed in tree structure, the respective computers are provided with correspondence list (tables) T0, T1 and T2 for recording the name of the other computer to be communicated with a certain computer, public key, time to use it and the name of a reliable computer, and when the public key is recognized, the information in the correspondence list is updated.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開平7－38555

(43)公開日　平成７年(1995)２月７日

(51)Int.Cl.[6]	識別記号	庁内整理番号	Ｆ Ｉ	技術表示箇所
Ｈ０４Ｌ　9/00				
Ｇ０９Ｃ　1/00		8837－5L		
			Ｈ０４Ｌ　9/00	

審査請求　未請求　請求項の数６　ＯＬ　（全 23 頁）

(21)出願番号　特願平5－155579

(22)出願日　平成５年(1993)６月25日

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号

(72)発明者　藤波　順久
東京都品川区東五反田３丁目14番13号　株式会社ソニーコンピュータサイエンス研究所内

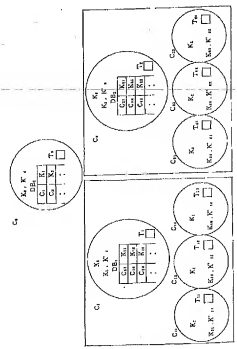
(74)代理人　弁理士　佐藤　隆久

(54)【発明の名称】　公開鍵暗号処理システムと方法

(57)【要約】

【目的】　公開鍵を用いて暗号処理を行って通信文を通信する公開鍵暗号処理システムにおいて、コンピュータの移動があっても、機密性および真正性を維持した公開鍵暗号処理方式を提供する。

【構成】　複数の通信媒体としてのコンピュータC_0，C_1，C_2が公開鍵K_0を用いて暗号処理した通信を行う公開鍵暗号処理システムにおいて、コンピュータで暗号に使用する鍵の管理が木構造で管理され、それぞれのコンピュータにそのコンピュータが通信を行う他のコンピュータの名称、公開鍵、使用された時刻、信頼したコンピュータの名称を記録する対応表（テーブル）T_0，T_1，T_2を設け、公開鍵を知ったとき、対応表の前記情報を更新する。



1

【特許請求の範囲】

【請求項１】公開鍵を用いて通信装置相互で暗号通信を行う公開鍵暗号処理システムであって、
それぞれが自己の公開鍵と該公開鍵に対応する秘密鍵を記憶している複数の第１の通信装置と、
前記複数の第１の通信装置をグループ分けし、それぞれのグループに属する前記第１の通信装置の名称とその公開鍵とを記憶し、さらに自己の公開鍵とそれに対応する秘密鍵を記憶し、そのグループに属する第１の通信装置の公開鍵の認証を行う第２の通信装置とを有し、
前記第２の通信装置のそれぞれには、その通信装置と通信可能な他の第２の通信装置が登録されており、
前記第１の通信装置および前記第２の通信装置のそれぞれに、それまでに公開鍵を知った前記第１または第２の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶する手段を有し、
第１または第２の通信装置を介して公開鍵を用いた通信が行われたとき、当該記憶手段の記憶内容を更新する公開鍵暗号処理システム。

【請求項２】前記第１の通信装置は、自己の前記記憶手段に、その通信装置が属するグループ内の他の第１の通信装置の公開鍵を事前に登録し、かつ、登録した通信装置について信頼する通信装置が存在しないと定義し、
該第１の通信装置は前記登録した第１の通信装置と、これらの通信装置が属する第２の通信装置を介して、直接公開鍵を用いて通信を行う、請求項１記載の公開鍵暗号処理システム。

【請求項３】第１の通信装置をあるグループから他のグループに変更したとき、
前のグループの第２の通信装置に記憶された、該削除した第１の通信装置の名称および公開鍵を削除し、
新たなグループの第２の通信装置に、該追加した第１の通信装置の名称および公開鍵を登録する請求項１または２記載の公開鍵暗号処理システム。

【請求項４】第１の通信装置をあるグループから削除したとき、そのグループの第２の通信装置に記憶された、該削除した第１の通信装置の名称および公開鍵を削除する、請求項１または２記載の公開鍵暗号処理システム。

【請求項５】第１の通信装置をあるグループに追加したとき、そのグループの第２の通信装置の前記記憶手段に、該追加した第１の通信装置の名称および公開鍵を登録する、請求項１または２記載の公開鍵暗号処理システム。

【請求項６】複数の通信装置が公開鍵を用いて暗号通信する公開鍵暗号処理方法であって、前記通信装置が使用する公開鍵の認証管理をグループ分け、かつ、階層化した木構造で行い、
それぞれの通信装置に自己の公開鍵とそれに対応する秘密鍵を記憶し、

2

前記階層の上位の通信装置において、自己の公開鍵とその秘密鍵に加えて、そのグループに属する下位の通信装置の名称とその公開鍵とを記憶し、
通信可能な上位の通信装置相互の関係を規定し、
前記通信装置のそれぞれは、それまでに公開鍵を知った通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶、更新する公開鍵暗号処理方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は公開鍵（Public-key encryption）処理システムに関するものである。特に、本発明は公開鍵暗号を用いて秘密通信や認証を行う大規模分散コンピュータ通信システムにおいて、公開鍵暗号のデータベースを分散管理する際に、通信先の公開鍵を知っているデータベースと直接公開鍵を交換することによってシステム内で信頼するべきマシン（コンピュータ）の交換数を減らすことが可能な公開鍵暗号処理システムに関する。

【０００２】

【従来の技術】たとえば、非常に多数、例えば１億台ものコンピュータを含むような世界規模のコンピュータネットワークシステムにおいて、各コンピュータは、システム内の他のコンピュータに通信文を送ることが可能なようにすることが試みられている。このような大規模コンピュータネットワークシステムにおいては、コンピュータとしては据え置き型のほか、移動可能なものも含まれており、コンピュータが移動しても通信文が相手のコンピュータに正確に到着するようにされている必要がある。このようなコンピュータネットワークシステムの各通信路では、悪意の送信者と受信者以外の者に通信内容を読み取られたり、通信内容を改変されたりする可能性がある。このような大規模分散コンピュータシステムは移動するオブジェクトやホストを含むため、実際の伝送に応じて通信相手を信頼できる程度が変化する。

【０００３】機密情報を通信するシステムにおいては、通信の秘密性（secrecy）と真正性（authenticity）を保証することが不可欠である。これまで通信の秘密性と真正性を保証する種々の暗号化方法と復号方法とが提案されている。たとえば、公開鍵暗号方式は、暗号化に使う鍵（公開鍵）と解読に使う鍵（秘密鍵）を異なるものとし、公開鍵から秘密鍵を推測し難くした暗号処理方式である。公開鍵暗号方式においては、公開鍵暗号方式を適用するそれぞれの各ユーザは固有の公開鍵と秘密鍵を持ち、二人のユーザは互いに相手の公開鍵を知るだけで秘密通信ができる。また逆に、秘密鍵を用いて変換した暗号文を、対応する公開鍵で読むことができる、そのユーザだけの作れる通信文となるため、ユーザの認証（本人確認）に使うこともできる。従って、公開鍵暗号を使って秘密通信や認証を行うには、ユーザの公開鍵の登録

された信頼できるデータベースがあればよい。そのため、データベース用の公開鍵を各ユーザにあらかじめ知らせておき、データベースからの情報は対応する秘密鍵で変換して送るようにする。

【０００４】大規模ネットワーク通信システムの場合、データベースの規模やアクセスの集中などの問題から、分散管理する必要がある。公開鍵を分散管理するシステムの例が、下記の文献に述べられている。

文献１：Butler Lampson, Martin Abadi, Michael Burrows, and Edward Wobber, "Authentication in Distributed Systems: Theory and Practice ", Proceedings of the 13th ACM Symposium on Operating System Principles, October 1991.

この文献に記載されているシステムでは、データベースは木構造状につながっており、公開鍵のデータベース自身の公開鍵は隣接するデータベースに含まれている。通信しようとする者が直接交信できるデータベースに通信相手の公開鍵がない場合、隣接するデータベースの公開鍵を順に得ていくことを、通信相手の公開鍵が含まれているデータベースに達するまで続ける。この場合、各データベースは一つ向こうのデータベースの公開鍵を正しく教えると仮定している。仮定を減らしたい場合、木構造以外の接続（cross link）数を増やす。

【０００５】また、共有鍵暗号をもとにした方法が下記文献に提案されている。

文献２：R. M. Needham and M. D. Schroeder: "Using Encryption for Authentication in Large Network of Computers", Communications of the ACM, Vol. 21, No. 12(1987), pp. 993-999.

このNeedham らの方法では、一つ、あるいは相互に信頼し合える認証サーバに各ホストとの共有鍵のデータベースを置く。そして、認証サーバは、要求があるごとにホストの対の間の通信のための共有鍵（セッションキー）を発行する。

【０００６】さらに、

文献３：Jennifer G. Steiner, Clifford Neuman, and Jeffery I. Schiller: Kerberos: "An Authentication Service for Open Network Systems", USENIX Winter Conference, USENIX Association, February 1988.

に提案したKerberosの方法はそのようなシステムの一例である。Kerborosシステムでも、認証の領域を複数用意し、他の領域に行くときにはその認証サーバのチケットをもらうようにすることができる。

【０００７】可搬型コンピュータの認証には、常に認証サーバにアクセスできるとは限らないという問題があり、これに関しては下記文献に記載がある。

文献４：岩井　三則、村田　賢一、所　真理雄：「可搬型計算機環境におけるホスト認証」，日本ソフトウェア科学会第９回大会予稿集，September 1992.

この文献に記載の方法では、通信する可能性のあるホストコンピュータとの共有鍵を暗号化したものをホストコンピュータが持ち、時々更新することで対処している。

【０００８】さらに上記文献１において、Lampson らは公開鍵暗号に基づいて認証を提案している。このLampson らの方法は公開鍵暗号に基づいた認証法を用いており、必要なのは単なる公開鍵のデータベースである。データベースは分散されており、データベース自身もその公開鍵によって認証される。

【０００９】

【発明が解決しようとする課題】大規模分散コンピュータネットワークシステムの場合、そのシステム内の通信を行う全ホストコンピュータの鍵を集中管理するのは記憶容量的にもトラフィック的にも不可能であり、分散管理が必要である。また、公開鍵暗号を使うとしても、秘密鍵の安全性の問題から、各鍵データベースのための鍵は、異なるものとしなければならない。つまり、鍵データベースの鍵の配送の問題が生じ、どの鍵データベースをどのように信用して用いたかが問題となる。文献１に示したLampson らの方法は、この問題を形式的に取り扱っているが、通信を行うホストコンピュータの移動は考慮していない。つまり、最近の移動可能な通信装置の進展に応じて、携帯性にすぐれたコンピュータを用いた通信装置をそのような大規模な通信システムに接続して一時的に通信システムに組み入れたり、外したりする運用が試みられており、そのような運用に公開鍵暗号を適用する場合に、公開鍵を入手するのに多大の経路をたどることは得策ではない。しかしながら、従来の方式においては、マシン（ここでは、通信を行うコンピュータ）の移動については考慮されていない。

【００１０】また、上述した大規模分散通信システムにおける公開鍵暗号処理においては、公開鍵を入手するまでに多大の経路をたどる場合がしばしば発生し、信頼する鍵を伝達するべきマシン（ここでは、正しい情報を教えると仮定したデータベース）の数を減らすには、新たな接続を増やさなければならない。特に、地球規模の大規模なコンピュータ通信システムにおいて公開鍵鍵暗号方式を適用する場合、公開鍵を正当に入手するまでに多大の経路をたどることになり、その経路において、信頼すべきマシンが多大になることは機密性および真正性の観点から好ましくない。これを改善するため、経路を少なくするように新たな接続を増加することは設備を変更するなどの観点から好ましくなく、現実的でもない。

【００１１】

【課題を解決するための手段】本発明では、このような環境で最大限の信頼性を実現するために、新たな接続を設けずに、ホストコンピュータの移動先で直接公開鍵を交換することで、通信において信頼できると仮定しなければならないホストコンピュータの数をなるべく少なくするようにする。

【００１２】したがって、本発明によれば、公開鍵を用

いて通信装置相互で暗号通信を行う公開鍵暗号処理システムであって、それぞれが自己の公開鍵と該公開鍵に対応する秘密鍵を記憶している複数の第１の通信装置と、前記複数の第１の通信装置をグループ分けし、それぞれのグループに属する前記第１の通信装置の名称とその公開鍵とを記憶し、さらに自己の公開鍵とそれに対応する秘密鍵を記憶し、そのグループに属する第１の通信装置の公開鍵の認証を行う第２の通信装置とを有し、前記第２の通信装置のそれぞれには、その通信装置と通信可能な他の第２の通信装置が登録されており、前記第１の通信装置および前記第２の通信装置のそれぞれに、それまでに公開鍵を知った前記第１または第２の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶する手段を有し、第１または第２の通信装置を介して公開鍵を用いた通信が行われたとき、当該記憶手段の記憶内容を更新する公開鍵暗号処理システムが提供される。好適には、上記通信装置には暗号コンピュータ通信を行うのに好適なコンピュータを含む。

【００１３】好適には、前記第１の通信装置は、自己の前記記憶手段に、その通信装置が属するグループ内の他の第１の通信装置の公開鍵を事前に登録し、かつ、登録した通信装置について信頼する通信装置が存在しないと定義し、該第１の通信装置は前記登録した第１の通信装置と、これらの通信装置が属する第２の通信装置を介して、直接公開鍵を用いて通信を行う。

【００１４】また、第１の通信装置をあるグループから他のグループに変更したとき、前のグループの第２の通信装置に記憶された、該削除した第１の通信装置の名称および公開鍵を削除し、新たなグループの第２の通信装置に、該追加した第１の通信装置の名称および公開鍵を登録する。

【００１５】あるいは、第１の通信装置をあるグループから削除したとき、そのグループの第２の通信装置に記憶された、該削除した第１の通信装置の名称および公開鍵を削除する。または、第１の通信装置をあるグループに追加したとき、そのグループの第２の通信装置の前記記憶手段に、該追加した第１の通信装置の名称および公開鍵を登録する。

【００１６】また本発明によれば、複数の通信装置が公開鍵を用いて暗号通信する公開鍵暗号処理方法であって、前記通信装置が使用する公開鍵の認証管理をグループ分け、かつ、階層化した木構造で行い、それぞれの通信装置に自己の公開鍵とそれに対応する秘密鍵を記憶し、前記階層の上位の通信装置において、自己の公開鍵とその秘密鍵に加えて、そのグループに属する下位の通信装置の名称とその公開鍵とを記憶し、通信可能な上位の通信装置相互の関係を規定し、前記通信装置のそれぞれは、それまでに公開鍵を知った通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶、更新する公開鍵暗号処理方法が提供される。

【００１７】

【作用】信用しなければならない通信装置（コンピュータ）の数を減らすには、通信を行う通信装置の近くに通信を行おうとする通信装置を持っていき、通信相手の公開鍵を登録し、認証を行う通信装置を経由せず、直接、相手の通信装置と通信を行う。つまり、大規模通信システムにおいても、小規模通信システムと同様に、直接、公開鍵をやりとりして、認証に関与する通信装置の数を減らして、暗号通信を行う。そのため、それぞれの通信装置に上述した記憶手段を設ける。

【００１８】特に、携帯型通信装置の場合、その通信装置の属するグループが頻繁に変更になる。その変更に応じて上記記憶手段の内容を変更し、好適には、新たに属するグループ内の通信を行う相手の通信装置の公開鍵を事前に登録して、その相手と直接、暗号通信を行う。この場合、認証に関与する通信装置は介在せず、機密性か，真正性の高い通信が可能となる。

【００１９】上述した記憶手段を設けると、通信装置の削除、追加に対しても、容易に対応可能となる。

【００２０】勿論、通信装置相互の通信は、大規模通信システムにおいても、可能である。この場合、公開鍵の管理をグループ化し、階層化した木構造をとることにより、信頼性の高い公開鍵暗号処理が可能となる。

【００２１】

【実施例】本発明の公開鍵暗号処理システムについて述べる。大規模分散コンピュータ通信システムにおいては、移動するオブジェクトやホストコンピュータを含むため、ホストコンピュータの実際の位置に応じて通信相手を信頼できる度合いが変化する。信用しなければならないホストコンピュータを少なくするためには、ホストコンピュータをその場に持って行って直接通信するのが確実である。遠くのホストコンピュータを知るにはいくつかのホストコンピュータを介するしかないわけで、認証においてもそれを信用しなければならないのは当然である。つまり、確実性を求めるならば、ホストコンピュータの移動を積極的に利用して、信用するべきホストコンピュータを減らせばよい。本発明では、従来問題となっていたホストコンピュータの移動を利用して、秘密通信や認証をなるべく少ないホストコンピュータを信用して通信する。本発明は、公開鍵データベースの管理に階層相対名前付け法の構造を用いている。階層相対名前付け法はマシンの移動に対応した名前付け法であり、本発明の公開鍵暗号処理システムの実施例として、移動可能なマシンを含む秘密通信・認証システムを例示する。

【００２２】まず、マイグレーションがない場合（ホストコンピュータの移動がない場合）の階層鍵管理方法について述べる。認証のためには相手を特定する必要があり、それは認証コードまたは名前（ＩＤ）で表される。

本発明においては、大規模分散システムに適したオブジェクトの名前付けとアドレシングの方法である、階層相対名前付け法をすでに下記文献に提案している。

文献５：藤波　順久、横手　靖彦：「大規模分散システムにおけるオブジェクトの名前づけ」、コンピュータソフトウェア、Vol.10, No.3(1993),pp.37-47.

この方法は、可搬型ホストコンピュータを含み、集中管理が不可能なほど大規模な分散コンピュータ通信システムにおいて、識別可能性、移動透過性、拡張性、高効率、可用性・耐故障性を持つオブジェクトの名前づけ法である。

【００２３】この方式では、仮定として、システムは論理的に階層構造をなしており、局所的名前空間を持っているとしている。大規模分散コンピュータ通信システム内のオブジェクトには、それが生成された名前空間（以下、「本籍」という）と現在いる名前空間（以下、「現住所」という）という概念がある。名前空間（のマネージャ）であるオブジェクトは、現在そこにいるオブジェクトと直接通信可能であるとする。通常、ホストコンピュータは一つの名前空間とそこにいるオブジェクトに対応しているため、ホストコンピュータの移動は名前空間の移動として手順が作られている。本発明でも、同様の仮定を用いて、階層鍵管理を行う。

【００２４】図１に示したように、各ホストコンピュータＡは自分の秘密鍵K_A^{-1}を保持しており、本籍のマネージャＭの公開鍵K_Mを知っている。また、本籍のマネージャＭはそれに属する全ホストコンピュータの公開鍵のデータベースを持ち、また、自分の秘密鍵K_M^{-1}を保持している。同じ本籍を持つホストコンピュータが通信している限り、鍵管理は単純である。すなわち、マネージャＭは認証コード（または名前）ＩＤ（Identification）で指定されたホストコンピュータの公開鍵をマネージャＭの秘密鍵でサインして返す。各ホストはマネージャＭの公開鍵を知っているので、公開鍵を取り出して相手ホストコンピュータを認証したり、秘密メッセージを送ったりできる。ホストコンピュータが異なるマネージャに属している場合には、いくつかのマネージャを順にたどっていって順番に公開鍵を得る必要がある。例えば、図１に示したホストコンピュータＡからホストコンピュータＤに秘密のメッセージを送るためにホストコンピュータＤの公開鍵を得たいとする。もしホストコンピュータＡが直接マネージャＴから公開鍵K_Dを送ってもらったとするとこれは認証されないので、間違った公開鍵が送られてきた可能性がある。

【００２５】そこで、下記手順、（１）ホストコンピュータＡはマネージャＳから鍵K_S $^{-1}$（K_R）を送ってもらい、鍵K_Sを使って読む、（２）ホストコンピュータＡはマネージャＲから鍵K_R $^{-1}$（K_T）を送ってもらい、鍵K_Mを使って読む、（３）ホストコンピュータＡはマネージャＴから鍵K_T $^{-1}$（K_D）を送ってもらい、鍵K_Dを使って読む、を経れば、安全に公開鍵K_Dを得ることができる。

【００２６】ここで、各マネージャは「一つ向こうの」公開鍵を正しく教えると仮定している。つまり、

（ａ）ホストコンピュータＡは、「鍵K_Sで読むと、ホストコンピュータＡ以外のマネージャＳに隣接するホストコンピュータが言ったことになっている命題」を、そのホストコンピュータが本当に言ったと信じる。

（ｂ）マネージャＳは、「鍵K_Rで読むと、マネージャＳ以外のマネージャＲに隣接するホストコンピュータが言ったことになっている命題」を、そのホストコンピュータが本当に言ったと信じる。

（ｃ）マネージャＲは、「鍵K_Tで読むと、マネージャＲ以外のマネージャＴに隣接するホストコンピュータが言ったことになっている命題」を、そのホストコンピュータが本当に言ったと信じる。

ということである。これとマネージャＴのデータベースの情報である、「ホストコンピュータＤは、自分の公開鍵がK_Dであると言った」という命題を組み合わせることで、ホストコンピュータＡはホストコンピュータＤの公開鍵がK_Dであると信じるようになる。このやり方は上記文献５で形式的に述べられている。

【００２７】上記仮定が成り立たなかった場合、つまり、どこかのマネージャが贋の公開鍵を返した場合、メッセージを相手が受け取ることができなくなってしまう。一方、マネージャは、メッセージの中継をごまかすことによってもメッセージを届かなくすることができる。つまり、マネージャが正しい公開鍵を返すかどうかは、マネージャを通るメッセージが正しく届けられるかどうかということと同程度に信用できる。換言すれば、従来の方法では、マネージャ（認証サーバ）は生成した共有鍵を用いてメッセージをこっそり盗聴することができ、これを検出するのは困難である。したがって、これは妥当な仮定である。

【００２８】次にマイグレーション（ホストコンピュータの移動）がある場合について述べる。ホストコンピュータが移動した場合、従来のやり方では、移動先でも本籍に認証してもらうか、または移動先でも認証してもらえるように本籍から手続きを行っておく必要があった。すると、ホストコンピュータの信用に関する仮定が増えてしまう。ところで、移動先で物理的接続がされるときには、そこのマネージャの公開鍵を直接入力することができる。また、同時にマネージャはホストコンピュータの公開鍵をデータベースに入れることができる。これを使うと、移動したホストコンピュータが移動先のホストコンピュータを認証する場合でも、その逆の場合でも、信用しなければならないホストコンピュータの数を減らすことができる。

【００２９】例えば図２でホストコンピュータＢが移動してマネージャＴと接続されたとする。そのときに、公

開鍵K_TをホストコンピュータBに入力し、また、マネージャTのデータベースに鍵K_Bを登録すれば、ホストコンピュータBは「鍵K_Tで読むと、ホストコンピュータB以外のマネージャTに隣接するホストコンピュータが言ったことになっている命題」を、そのホストコンピュータが本当に言ったと信じるという仮定のもとで、そのホストコンピュータ、例えばホストコンピュータCの認証ができるようになる。したがって、上述した仮定（ａ）～（ｃ）は不要である。

【００３０】逆にいえば、移動先でホストコンピュータが特に公開鍵を直接入力することがなければ、本籍のマネージャから始めて今までと同じ仮定をおく必要があるし、相手がこちらを認証するには移動先のマネージャは本籍のマネージャに頼んで（やはり同様の仮定が必要）公開鍵を取り寄せなければならない。

【００３１】階層相対名前付け法におけるホストの移動手順に認証を付け加えた例を述べる。階層相対名前付け法では、本籍は常にそれに属するオブジェクトの現在位置を正しく知っている必要があるため、切断通知、新住所通知、確認通知には認証が加わった。この例では、移動するホストコンピュータをＡとする。

【００３２】（１）移動開始：ホストコンピュータＡは本籍のマネージャに切断通知を送る。これには、ホストコンピュータＡの秘密鍵でサインしたオブジェクト認証コードＯＩＤ（Object ID）とオブジェクトアドレスＯＡＤ（Object address）とを付け加えて認証する。このオブジェクトアドレスＯＡＤにはタイムスタンプが含まれているため、切断通知の再送は防止される。このメッセージが通過した名前空間のマネージャは、ホストＡとその子孫に対する局部認証コードＬＩＤ（Local ID）とオブジェクトアドレスＯＡＤ、オブジェクト認証コードＯＩＤと局部アドレスＬＡＤ（Local Address）、または、オブジェクト認証コードＯＩＤとオブジェクトアドレスＯＡＤの組を無効にする（これは認証されなくてもよい）。

【００３３】（２）移動：ホストコンピュータＡが移動する。

【００３４】（３）移動終了：新しい現住所のマネージャに局部アドレスＬＡＤを割り当ててもらう。マネージャはこのオブジェクト認証コードＯＩＤと局部アドレスＬＡＤの組を記憶する。このとき同時に、ホストコンピュータＡは現住所のマネージャの公開鍵を記憶し、また、ホストコンピュータＡの公開鍵を現住所のマネージャのデータベースに登録することが望ましい。そして、ホストコンピュータＡは、本籍のマネージャにオブジェクト認証コードＯＩＤ、新しいオブジェクトアドレスＯＡＤ、仮想的なオブジェクト認証コードＯＩＤである（０：）と、それらをホストコンピュータＡの秘密鍵でサインしたものを通知する。本籍のマネージャはオブジェクトアドレスＯＡＤを更新し、確認通知を返す。確認通知には、新住所通知のオブジェクトアドレスＯＡＤについていたタイムスタンプと、仮想的なオブジェクト認証コードＯＩＤＯ：の現在の値（逆ＯＩＤ）が、本籍の秘密鍵でサインされたものが含まれている。

【００３５】階層相対名前付け法の場合、認証コードＩＤが相対表現なので、本籍のマネージャと認証通信して相対位置を確認しないと認証コードＩＤが決められない。これはつまり、認証コードＩＤの扱いに関しては途中のホストコンピュータを信用しているということである。これは次の２つの点で問題がある。もし本籍のマネージャと通信できなかった場合どことも通信が始められないこと、および、依然として信用すべきホストコンピュータが減らないことである。このうち特に前者を解決する方法として、本籍のバックアップの働きをするホストコンピュータを用意する。後者の問題は、バックアップの認証コードＩＤの確定の問題があって解決し難いが、ホストコンピュータの移動前に移動先を決めておくことによってある程度解決できる。

【００３６】オブジェクトについてマイグレーションが起きる場合は、ホストコンピュータについてと同様に、オブジェクトも認証の対象にしなければならない。ただし、移動先のホストコンピュータは信用できなければならない。ホストコンピュータはオブジェクトの全データに（秘密鍵にも）アクセスできるからである。機密情報を交信するという点では、通常の通信もオブジェクトマイグレーションも相手のホストコンピュータを同程度に信用している必要がある。ただ、外から見た場合、移動先のホストコンピュータを認証するよりもオブジェクトそのものを認証できたほうが都合がよいこと、移動先からの新住所通知をオブジェクトの鍵で認証できれば便利なことから、オブジェクトにも秘密鍵と公開鍵の組を割り当てる。

【００３７】オブジェクトマイグレーションの手順は、ホストコンピュータの場合とほとんど同じである。オブジェクトが初めて本籍を離れるときに、マネージャがオブジェクトに秘密鍵を割り当て、公開鍵をデータベースに登録するという点が異なる。オブジェクトが本籍に帰るときには、公開鍵と対応する秘密鍵を破棄し、次回は別の鍵を使うこともできる。

【００３８】本発明の方法では、従来の方法に比べると、移動した先にあるホストコンピュータとの通信で用いる仮定が少なくなり、より信頼できる通信が可能である。さらに、移動先のデータベースの公開鍵を覚えたままにしておけば、ホストコンピュータが元の場所に戻ってからも、この信頼性は変わらない。

【００３９】一方、本発明の方法の欠点として、前述した認証コードＩＤの確定の問題の他に、鍵の更新の問題が予想される。つまり、ホストコンピュータが公開鍵を更新しようとしたとき、本籍と通信ができない可能性がある。本籍のバックアップを用意して解決しようとした

場合、さらにバックアップ同士の整合性の問題が発生してしまう。一つの解決策としては、ホストコンピュータは通常の秘密鍵の他にマスター秘密鍵を持っていて、それに対する公開鍵は本籍のマネージャが記憶しているとして、鍵の更新はマスター秘密鍵を使ってサインすることが考えられる。同じ鍵を使って送る情報の量を少なくするという点では、安全性に貢献できる方法と言える。また、一般に公開鍵暗号による通信は、共有鍵の場合に比べて非常に遅いという問題があるが、これについては、共有鍵を公開鍵暗号を用いて交換する方法を用いればよい。

【００４０】以上に述べたように、本発明に基づく大規模分散システムに適した公開鍵認証のための鍵管理においては、鍵は階層管理され、鍵データベースへのアクセスの集中が防止でき、また、ホストコンピュータの移動を積極的に利用して、信用しなければならないホストコンピュータの数をできるだけ少なくした秘密・認証通信ができる。

【００４１】上述した本発明について、さらに詳細について述べる。まず、本発明の基本技術である公開鍵鍵暗号方式について述べる。平文とは、暗号化される前のデータをいい、暗号文とは、暗号化後のデータをいう。暗号変換式E_Kは平文から暗号文への鍵Kによって決まる変換式であり、復号変換式$D_{K'}$は、暗号文から平文への、鍵K'によって決まる変換式である。鍵から暗号化変換式、復号化変換式を決める手順は公開されているとする。各平文Mについて、

【数１】

$$D_{K'}(E_K(M))=M$$

である。

【００４２】ＤＥＳ（Data Encryption Standard, FIPS PUB 46, National Bureau of Standards, Washington, D.C. (Jan. 1977)）などの共有鍵暗号では、K＝K'である。1976年にDiffie とHellman によって紹介された下記文献、

文献６：W. Diffie and M. Hellman, "New Direction in Cryptography ", IEEE Transactions on Infomation Theory Vol. IT-22(6) pp. 644-654 、Nov1976

に記載された公開鍵暗号方式では、鍵Kと鍵K'とは異なり、鍵Kから鍵K'を求めることは非常に困難である。

【００４３】公開鍵暗号方式は、上述したように、暗号化に使う鍵（公開鍵）と解読に使う鍵（秘密鍵）を異なるものとし公開鍵から秘密鍵を推測し難くした暗号処理方式であり、この公開鍵暗号を用いて秘密通信を行うシステムでは、各ユーザ（ホストコンピュータ）Aは２つの鍵K_Aと鍵K_A'を持っている。鍵K_Aは公開鍵と呼ばれ、公開鍵データベースに登録されている。公開鍵データベースは、ユーザ名Aを指定するとその公開鍵K_Aを答える機能を持つ。鍵K_A'は秘密鍵と呼ばれ、ユーザAのみが知っている。ユーザAが平文Mを秘密裡にユーザBに送ろうとするときには、ユーザAは公開鍵データベースから鍵K_Bを得、ユーザBに暗号文

【数２】

$$C=E_{K_B}(M)$$

を送る。ユーザBは、それを受信して平文

【数３】

$$M=D_{K_B'}(C)$$

を得る。秘密鍵K_B'はユーザBのみが知っており、しかも、鍵K_Bから鍵K_B'を計算するのは非常に困難であるため、ユーザB以外のユーザが平文Mを得るのを防ぐことができる。

【００４４】公開鍵暗号を用いて認証（本人確認）を行うシステムでは、暗号化変換と復号化変換に対する仮定として、平文を復号化変換できること、暗号文を暗号化変換できること、各平文Mについて

【数４】

$$M=E_K(D_{K'}(M))$$

であることを要請する。上と同様に各ユーザが二つの鍵を持ち、公開鍵データベースを用意するならば、ユーザAが平文MをユーザBに、確かにユーザAから送られたことがわかるように送るときには、ユーザAはユーザBに

【数５】

$$C=D_{K_A'}(M)$$

を送る。ユーザBへはそれを受信して後、あるいはあらかじめ公開鍵データベースから鍵K_Aを得、それを用いて平文

【数６】

$$M=E_{K_A}(C)$$

を得る。秘密鍵K_A^{-1}はユーザAのみが知っており、しかも、鍵K_Aから鍵K_A'を計算するのは非常に困難であるため、興味のある平文に復元できるようなコードを作れるのはユーザAだけである。つまり、ユーザA以外のユーザがユーザAのふりをして通信することはできない。

【００４５】公開鍵暗号の具体例としては、

文献７：R. L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems ", Communications ofthe ACM Vol 21(2) pp. 120-126 、 Feb. 1978

に記載されたＲＡＳ方式が知られている。

【００４６】非常に多くのコンピュータを含むネットワークシステムの場合、ユーザ数も非常に多く、公開鍵データベースをシステム内に１個だけ用意しておいた場合、データベースは巨大になり、また、データベースへのアクセス頻度が非常に高くなる。これを避けるために公開鍵データベースは分散管理するのが普通である。例

えば、上述した文献1（Butler Lampson, Martin Abadi, Michael Burrows, and Edward Wobber, "Authentication in Distributed Systems: Theory and Practice", Proceedings of the 13th ACM Symposium on Operating System Principles, October 1991）に記載されたLampsonらのシステムの一例を次に示す。

【0047】図3で、記号C_0, C_1, C_2, …C_{11}, C_{12}, …C_{21}, C_{22}, …は通信を行うコンピュータ（計算機）を示しており、鍵K，K'から暗号化変換式E_Kと復号化変換式$D_{K'}$とを求め、それを実行することができる。ここでは、ネットワークシステムに接続されているコンピュータが3段の木構造に管理されているとする。つまり、記号C_1はコンピュータC_1, C_{11}, C_{12}, …を代表するコンピュータを示しており、コンピュータC_{11}, C_{12}, …の公開鍵K_{11}, K_{12}, …を記憶している公開鍵データベースD_{B1}を持っている。同様に、コンピュータC_2, はコンピュータC_1, C_{21} C_{22}, …を代表するコンピュータであり、コンピュータC_{21}, C_{22}, …の公開鍵K_{21}, K_{22}, …を記憶している公開鍵データベースD_{B2}を持っている。また、コンピュータC_0はコンピュータC_1, C_2, …を代表するコンピュータであり、コンピュータC_1, C_2, …の公開鍵K_1, K_2, …を記憶している公開鍵データベースD_{B0}を持っている。各コンピュータは一人のユーザが使用しているとする。つまり、各コンピュータC_nは公開鍵K_nと秘密鍵K_n'を記憶している。公開鍵を公開鍵データベースに登録する作業は、人手で本人確認をして行う。

【0048】このシステムでは、公開鍵データベースから公開鍵を得るときにもネットワークを使用する。従って、公開鍵データベースを持つコンピュータ以外から鍵の公開鍵を与えられることを防ぐために、公開鍵データベースも認証の対象にする。例えば、コンピュータC_{11}には、自分の公開鍵と秘密鍵のほかに公開鍵データベースD_{B1}を持つコンピュータC_1の公開鍵K_1が予め人手で入力されているる。コンピュータC_{12}の公開鍵を得たい場合、「コンピュータC_{12}の公開鍵はK_{12}である。」という通信文をM_{12}とすれば、コンピュータC_1からコンピュータC_{11}に復号文D_{K1}'（M_{12}）を送ってもらえばよい。コンピュータC_{11}は公開鍵K_1を用いて通信文$M_{12}=E_{K1}(D_{K1}'(M_{12}))$を得、確かにコンピュータ$C_1$から送られたものと確認することができる。ここでは、コンピュータC_1は十分に信頼でき、コンピュータC_{11}, C_{12}, …の公開鍵K_{11}, K_{12}, …を正しく教えると仮定している。

【0049】代表となるコンピュータが異なるようなコンピュータの公開鍵を得ようとする場合は、いくつかの公開鍵データベースを順に検索していく必要がある。例えば、コンピュータC_{11}がC_{21}の公開鍵を得たい場合、「コンピュータC_0の公開鍵はK_0である。」という通信文をM_0とすれば、まずコンピュータC_1に頼んでD_{K1}'（M_0）を送ってもらう。コンピュータC_{11}は公開鍵K_1を用いて$M_0=E_{K1}(D_{K1}(M_0))$を得る。次に、「コンピュータC_2の公開鍵はK_2である。」という通信文をM_2とするとき、コンピュータC_0に頼んで$D_{K0}(M_2)$を送ってもらう。コンピュータC_{11}は先ほど得た公開鍵K_0を用いて$M_{21}=E_{K2}(D_{K2}(M_{21}))$を得る。最後に、「コンピュータ$C_{21}$の公開鍵は$K_{21}$である。」という通信文を$M_{21}$とするとき、コンピュータ$C_2$に頼んで$D_{K2}'$（$M_{21}$）を送ってもらう。コンピュータ$C_{11}$は先ほど得た公開鍵$K_2$を用いて$M_{21}=E_{K2}(D_{K2}'(M_{21}))$を得る。こうしてコンピュータ$C_{21}$の公開鍵$K_{21}$を得ることができる。ここでは、コンピュータ$C_1$, C_0, C_2, は十分に信頼できる、その一つ向こうのコンピュータの公開鍵を正しく教えると仮定している。

【0050】信頼するべき経路を辿るコンピュータの数を減らしたい場合には、公開鍵をあらかじめ知らせてあるコンピュータを、木構造で管理されている階層より抑やせばよい。例えば、コンピュータC_2の公開鍵K_2をあらかじめ人手でコンピュータC_1に入力しておけば、コンピュータC_0を信頼するという仮定は必要なくなる。ただし、コンピュータC_1がコンピュータC_2の公開鍵を知っているという情報は、何らかの形でコンピュータC_{11}が知る必要がある。

【0051】文献1に示したLampsonらのシステムの問題点は、上述したように移動可能なコンピュータについて考慮されていないことである。例えば、コンピュータC_{11}が移動可能な計算機であって、コンピュータC_{11}のユーザがコンピュータC_{11}をコンピュータC_{21}の近くに持って行ったとする。そのときでもコンピュータC_{11}からコンピュータC_{21}に秘密の通信文を送ろうとするときには、コンピュータC_1, C_0, C_2 と順に通信してコンピュータC_{21}の公開鍵K_{21}を得なければならないし、コンピュータC_1, C_0, C_2 は信頼できると仮定しなければならない。逆に、コンピュータC_{21}からコンピュータC_{11}に秘密の通信文を送ろうとするときにも、コンピュータC_2, C_0, C_1 と順に通信してコンピュータC_{11}の公開鍵K_{11}を得なければならないし、コンピュータC_2, C_0, C_1 は信頼できると仮定しなければならない。

【0052】本発明では、上の問題を解決するために、各コンピュータはそれまでに公開鍵を知ったコンピュータの名称とその公開鍵、そしてそれを知るために信頼したコンピュータの名称の対応表（テーブル）を持っているとする。対応表は公開鍵データベースから公開鍵を教えてもらったときのほか、コンピュータのユーザがコンピュータの名称とその公開鍵を入力したとき（このときには信頼したコンピュータの名称は空である）にも更新される。特に、移動可能なコンピュータの場合には、他のコンピュータの近くに移動したときに、そのコンピュータのユーザから直接公開鍵を教えてもらえることが期

待できるので、信頼したコンピュータの数の少ない公開鍵を得ることができる。

【００５３】例えば、図４でコンピュータC_{11}のユーザがコンピュータC_{11}をコンピュータC_{21}の近くに持って行ったときに、コンピュータC_{21}のユーザからその公開鍵K_{21}を教えてもらったとする。すると、他のコンピュータを信頼することなく、コンピュータC_{11}からコンピュータC_{21}に秘密の通信文を送ることができる。この状況は、コンピュータC_{11}を元の場所に持って帰ったり、さらに他の場所に移動したりしてからでも変わらない。また逆に、コンピュータC_{11}のユーザがコンピュータC_{21}のユーザに公開鍵K_{11}を教えれば、他のコンピュータを信頼することなく、C_{21}からC_{11}に秘密の通信文を送ることができるし、コンピュータC_{11}をまた移動してからでも同様である。

【００５４】他の例として、コンピュータC_{11}をコンピュータC_2の近くに持って行き、公開鍵K_2を入力したとする。すると、コンピュータC_{22}に秘密の通信文を送りたい場合、すでに対応表に、「コンピュータC_2の公開鍵はK_2で、信頼したコンピュータはない」という情報があるため、コンピュータC_2と通信して公開鍵K_{22}を教えてもらうだけでよい。この場合、信頼したコンピュータはコンピュータC_2のみである。このようにして、公開鍵を直接入力したコンピュータを中心としたいくつかのコンピュータについては、その公開鍵を得るために信頼したコンピュータの数を、Lampsonらのシステムに比べて少なくすることができる。

【００５５】公開鍵を得たいコンピュータの名前と、すでに対応表に記録されているコンピュータの名前から、どのコンピュータと順に通信していくかを決定するために、コンピュータの名前付けはコンピュータの管理の木構造、つまり、鍵管理の木構造を反映したものでなければならない。これは、Lampsonらのシステムでも同様である。

【００５６】図４で、コンピュータC_0，C_1，C_2，…は据え置き型コンピュータであり、C_{11}，C_{12}，…C_{21}，C_{22}，…は据え置き型、あるいは移動可能なコンピュータである。各コンピュータは、鍵K，K'から暗号化変換式E_kと復号化変換式D_k'を求め、それを実行することができる。これらのコンピュータはコンピュータネットワークシステムに含まれており、各コンピュータは、システム内の他のコンピュータに通信文を送ることができる。コンピュータが移動しても通信文は相手のコンピュータに到着するものとする。ネットワークシステムの各通信路では、通信文の送信者と受信者以外の者に通信内容を読み取られたり、通信内容を改変されたりする可能性がある。

【００５７】コンピュータネットワークシステムに接続されているコンピュータは、鍵管理の３段の木構造に管理されているとする。つまり、コンピュータC_1はコンピュータC_1，C_{11}，C_{12}，…を代表するコンピュータであり、コンピュータC_{11}，C_{12}，…の公開鍵K_{11}，K_{12}，…を記憶している公開鍵データベースDB_1を持っている。同様に、コンピュータC_2はコンピュータC_1，C_{21}，C_{22}，…を代表するコンピュータであり、コンピュータC_{21}，C_{22}，…の公開鍵K_{21}，K_{22}，を記憶している公開鍵データベースDB_2を持っている。また、コンピュータC_0はコンピュータC_1，C_2…を代表するコンピュータであり、コンピュータC_1，C_2…の公開鍵K_1，K_2，…を記憶している公開鍵データベースDB_0を持っている。各コンピュータは一人のユーザが使用しているとする。つまり、各コンピュータC_nは公開鍵K_nと秘密鍵K_n'を記憶している。公開鍵と公開鍵データベースに登録する作業は、人手で本人確認をして行う。

【００５８】各データベースが管理しているコンピュータと鍵の数はたとえば、１０００～１００００程度とする。公開鍵暗号としてＲＳＡ方式を用いた場合、公開鍵は２００桁の１０進数２個（１．３キロビット）程度の記憶領域を必要とするため、データベースの大きさは１．３～１３メガビット程度となる。

【００５９】各コンピュータC_0，C_1，C_2，（一般的にはC_nとして表す）は、表（テーブル）T_0，T_1，T_2，（一般的にはT_nとして表す）を持つ。対応表の各行には、コンピュータの名前（名称）、公開鍵、使われた時刻、信頼したコンピュータの名称（集合）が記録されている。対応表の行数は１００～１０００程度とし、それを越える行を記憶させようとした場合には、信頼したコンピュータの名称の数が多い順、数が同じものが複数あるときには時刻の古い順に消去する。つまり、新しいものを残しておく。コンピュータの名称の数の大きさなどを無視すれば、対応表の大きさは１３０キロビット～１．３メガビット程度となる。

【００６０】コンピュータC_nのユーザがコンピュータC_pの公開鍵K_pを入力したときの表更新手順を図５に示す。まず、対応表T_nにコンピュータC_pについての行があるとき（ステップ１）は、コンピュータC_pについての行を作成する（ステップ２）。ないときには、その行の公開鍵とK_pを比べる（ステップ３）。異なる場合には、信頼したコンピュータのうちのどれかが嘘をついたことがわかるため、その旨エラー表示した後（ステップ４）、信頼したコンピュータの名称にコンピュータC_pを含む、対応表T_nのすべての行を削除する（ステップ５）。一致したなら、その行の信頼したコンピュータの名称をC，C' C''…とするとき（ステップ６）、対応表T_nの各行のうち、信頼したコンピュータの名称にC_p，C，C'，C''…が含まれるなら、コンピュータC，C'，C''…を除く（ステップ７）。最後にコンピュータC_pについての行の公開鍵にK_pを、使われた時刻に現在時刻を、信頼した計算機名に空集合を設定する（ステップ８）。

【００６１】コンピュータC_nのユーザがC_pの公開鍵を得ようとしたときの表更新手順を図６に示す。まず、コンピュータC_pがコンピュータC_nの代表となるコンピュータであった場合は、すでに公開鍵を知っているので終了する（ステップ１１）。対応表T_nにコンピュータC_pについての行があるとき（ステップ１２）はそれを使えばよいので終了する。ないときには、対応表T_nの各行と代表となるコンピュータについて、コンピュータC_pから木構造をたどったときの段数と信頼したコンピュータの数（代表となるコンピュータの場合は０）の和を計算し（ステップ１３）、和の最も少ないコンピュータをC_qとする（ステップ１４）。コンピュータの数が複数あるときは段数の少ないものから任意に一つを選ぶ。コンピュータC_qに木構造で隣接し、コンピュータC_pに一段近いコンピュータ（一意に決まる）をコンピュータC_rとする（ステップ１５）。対応表T_nのコンピュータC_qについての行の使われた時刻を現在時刻とする（ステップ１６）。コンピュータC_qと通信してコンピュータC_rの公開鍵K_rを得る（ステップ１７）。コンピュータC_qについての行の信頼したコンピュータの名称をC，C'，C''…とする（ステップ１８）。対応表T_nにコンピュータC_rについての行を作成し（ステップ１９）、公開鍵にK_rを、使われた時刻に現在時刻を、信頼したコンピュータの名称にコンピュータC，C'，C''…を設定する（ステップ２０）。そしてステップ１２に戻る。

【００６２】以上のように処理することにより、ホストコンピュータの移動、あるいは、オブジェクトの移動があっても、大規模分散コンピュータ通信ネットワークシステムにおいて、信頼するコンピュータの数を減らして信頼性の高い通信を行うことができる。

【００６３】

【発明の効果】本発明によれば、ユーザが望むならば直接公開鍵をやりとりすることで、小規模なシステムと同様の信頼性を得ることができる。また、それほど信頼性を要求しない場合は、間接的に公開鍵を得ることもできき、ユーザの要求レベルに合わせた信頼性で秘密通信・認証が可能になる。

【００６４】また本発明に基づく大規模分散システムに適した公開鍵認証のための鍵管理においては、鍵は階層管理され、鍵データベースへのアクセスの集中が防止でき、また、ホストコンピュータの移動を積極的に利用して、信用しなければならないホストコンピュータの数をできるだけ少なくした秘密・認証通信ができる。

【００６５】さらに本発明の方式は、従来の方式に比べると、移動した先にあるホストコンピュータとの通信で用いる仮定が少なくなり、より信頼できる通信が可能である。さらに、移動先のデータベースの公開鍵を覚えたままにしておけば、ホストコンピュータが元の場所に戻ってからも、この信頼性は変わらない。

【図面の簡単な説明】

【図１】従来の公開鍵暗号処理システムの構成図である。

【図２】本発明の公開鍵暗号処理システムの構成図である。

【図３】従来の公開鍵暗号処理システムの構成図である。

【図４】本発明の公開鍵暗号処理システムの構成図である。

【図５】図４に示した公開鍵暗号処理システムにおける公開鍵を直接入力する場合の処理方法を示すフローチャートである。

【図６】図４に示した公開鍵暗号処理システムにおける公開鍵を入手するときの更新手順を示すフローチャートである。

【符号の説明】

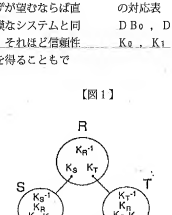
C_0，C_1，C_2，C_n・・通信を行うコンピュータ

T_0，T_1，T_2，T_n・・通信を行うコンピュータ内の対応表

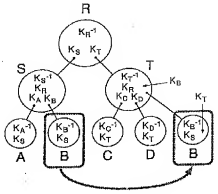
DB_0，DB_1，DB_2，DB_n・・データベース

K_0，K_1，K_2，K_n・・鍵

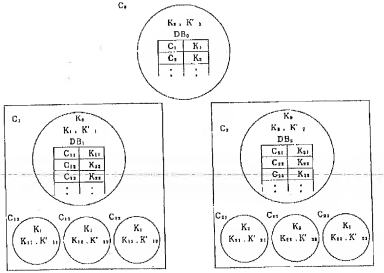
【図１】



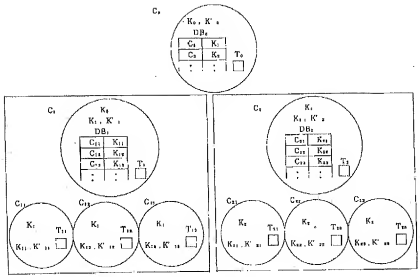
【図2】



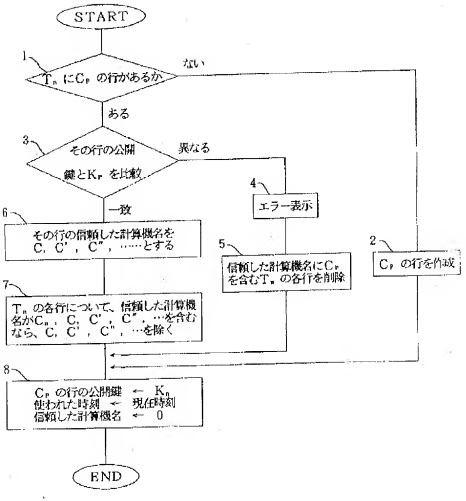
【図3】



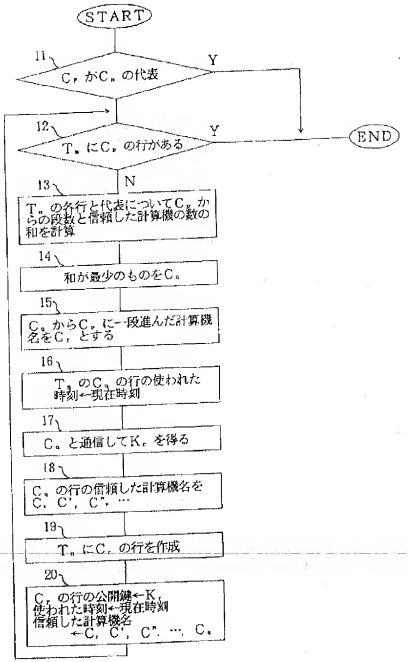
【図4】


C0
K0 , K' 0
UB0
C01 K01
C02 K02
C03 K03
T0
C1
K1
K1 , K' 1
DB1
C11 K11
C12 K12
C13 K13
T1
C2
K2
K2 , K' 2
DB2
C21 K21
C22 K22
C23 K23
T2
C11
K11
T11
K11 , K' 11
C12
K12
T12
K12 , K' 12
C13
K13
T13
K13 , K' 13
C21
K21
T21
K21 , K' 21
C22
K22
T22
K22 , K' 22
C23
K23
T23
K23 , K' 23

【図5】


START
1
T_n に C_P の行があるか
ない
ある
3
その行の公開鍵と K_P を比較
異なる
一致
4
エラー表示
6
その行の信頼した計算機名を C, C', C", ……とする
5
信頼した計算機名に C_P を含む T_n の各行を削除
2
C_P の行を作成
7
T_n の各行について、信頼した計算機名が C_n, C, C', C", …を含むなら、C, C', C", …を除く
8
C_P の行の公開鍵 ← K_n
使われた時刻 ← 現在時刻
信頼した計算機名 ← 0
END

【図6】



【手続補正書】
【提出日】平成6年5月11日
【手続補正1】
【補正対象書類名】明細書
【補正対象項目名】全文
【補正方法】変更
【補正内容】
【書類名】　明細書
【発明の名称】　公開鍵暗号処理システムと方法
【特許請求の範囲】
【請求項1】公開鍵を用いて通信装置相互で暗号通信を

行う公開鍵暗号処理システムであって、
それぞれが自己の公開鍵と該公開鍵に対応する秘密鍵を記憶している複数の第１の通信装置と、
前記複数の第１の通信装置をグループ分けし、それぞれのグループに属する前記第１の通信装置の名称とその公開鍵とを記憶し、さらに自己の公開鍵とそれに対応する秘密鍵を記憶し、そのグループに属する第１の通信装置の公開鍵の認証を行う第２の通信装置とを有し、
前記第２の通信装置のそれぞれには、その通信装置と通信可能な他の第２の通信装置が登録されており、
前記第１の通信装置および前記第２の通信装置のそれぞれに、それまでに公開鍵を知った前記第１または第２の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶する手段を有し、
通信装置が他の通信装置の公開鍵を入手したとき、当該記憶手段の記憶内容を更新する公開鍵暗号処理システム。
【請求項２】前記第１の通信装置は、自己の前記記憶手段に、その通信装置が属するグループ内の他の第１の通信装置の公開鍵を事前に登録し、かつ、登録した通信装置について信頼する通信装置が存在しないと定義し、
該第１の通信装置は前記登録した第１の通信装置と、直接公開鍵を用いて通信を行う、請求項１記載の公開鍵暗号処理システム。
【請求項３】第１の通信装置をあるグループから他のグループに一時的に変更したとき、
新たなグループの第２の通信装置の前記記憶手段に、該追加した第１の通信装置の名称および公開鍵を登録する請求項１または２記載の公開鍵暗号処理システム。
【請求項４】第１の通信装置をあるグループに追加したとき、そのグループの第２の通信装置に、該追加した第１の通信装置の名称および公開鍵を登録する、請求項１または２記載の公開鍵暗号処理システム。
【請求項５】複数の通信装置が公開鍵を用いて暗号通信する公開鍵暗号処理方法であって、前記通信装置が使用する公開鍵の認証管理をグループ分け、かつ、階層化した木構造で行い、
それぞれの通信装置に自己の公開鍵とそれに対応する秘密鍵を記憶し、
前記階層の上位の通信装置において、自己の公開鍵とその秘密鍵に加えて、そのグループに属する下位の通信装置の名称とその公開鍵とを記憶し、
通信可能な上位の通信装置相互の関係を規定し、
前記通信装置のそれぞれは、それまでに公開鍵を知った通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶、更新する公開鍵暗号処理方法。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】本発明は公開鍵暗号（Public-key encryption）処理システムに関するものである。特に、本発明は公開鍵暗号を用いて秘密通信や認証を行う大規模分散コンピュータ通信システムにおいて、公開鍵のデータベースを分散管理する際に、通信先の公開鍵を知っているデータベースと直接公開鍵を交換することによってシステム内で信頼するべきマシン（コンピュータ）の交換数を減らすことが可能な公開鍵暗号処理システムに関する。
【０００２】
【従来の技術】たとえば、非常に多数、例えば１億台ものコンピュータを含むような世界規模のコンピュータネットワークシステムにおいて、各コンピュータは、システム内の他のコンピュータに通信文を送ることが可能なようにすることが試みられている。このような大規模コンピュータネットワークシステムにおいては、コンピュータとしては据え置き型のほか、移動可能なものも含まれており、コンピュータが移動しても通信文が相手のコンピュータに正確に到着するようにされている必要がある。このようなコンピュータネットワークシステムの各通信路では、通信文の送信者と受信者以外の者に通信内容を読み取られたり、通信内容を改変されたりする可能性がある。このような大規模分散コンピュータシステムは移動するオブジェクトやホストを含むため、実際の位置に応じて通信相手を信頼できる程度が変化する。
【０００３】機密情報を通信するシステムにおいては、通信の秘密性(secrecy)と真正性(authenticity)を保証することが不可欠である。これまで通信の秘密性と真正性を保証する種々の暗号化方法と復号方法とが提案されている。たとえば、公開鍵暗号方式は、暗号化に使う鍵（公開鍵）と解読に使う鍵（秘密鍵）を異なるものとし、公開鍵から秘密鍵を推測し難くした暗号処理方式である。公開鍵暗号方式においては、公開鍵暗号方式を適用するそれぞれの各ユーザは固有の公開鍵と秘密鍵を持ち、二人のユーザは互いに相手の公開鍵を知るだけで秘密通信ができる。また逆に、秘密鍵を用いて変換した暗号文は、対応する公開鍵で読むことができる、そのユーザだけの作れる通信文となるため、ユーザの認証（本人確認）に使うこともできる。従って、公開鍵暗号を使って秘密通信や認証を行うには、ユーザの公開鍵の登録された信頼できるデータベースがあればよい。そのため、データベース用の公開鍵を各ユーザにあらかじめ知らせておき、データベースからの情報は対応する秘密鍵で変換して送るようにする。
【０００４】大規模コンピュータネットワーク通信システムの場合、データベースの規模やアクセスの集中などの問題から、分散管理する必要がある。公開鍵を分散管理するシステムの例が、下記の文献に述べられている。
文献１：Butler Lampson, Martin Abadi, Michael Burrows, and Edward Wobber, "Authentication in Distri

buted Systems: Theory and Practice", Proceedings of the 13th ACM Symposium on Operating System Principles, October 1991.
この文献に記載されているシステムでは、データベースは木構造状につながっており、公開鍵のデータベース自身の公開鍵は隣接するデータベースに含まれている。通信しようとする者が直接交信できるデータベースに通信相手の公開鍵がない場合、隣接するデータベースの公開鍵を順に得ていくことを、通信相手の公開鍵が含まれているデータベースに達するまで続ける。この場合、各データベースは一つ向こうのデータベースの公開鍵を正しく教えると仮定している。仮定を減らしたい場合、木構造以外の接続（cross link）数を増やす。

【０００５】また、共有鍵暗号をもとにした方法が下記文献に提案されている。

文献２：R. M. Needham and M. D. Schroeder: "Using Encryption for Authentication in Large Network of Computers", Communications of the ACM, Vol. 21, No. 12(1978), pp. 993-999.

このNeedham らの方法では、一つ、あるいは相互に信頼し合える認証サーバに各ホストとの共有鍵のデータベースを置く。そして、認証サーバは、要求があるごとにホストの対の間の通信のための共有鍵（セッションキー）を発行する。

【０００６】さらに、

文献３：Jennifer G. Steiner, Clifford Neuman, and Jeffery I. Schiller: Kerberos: "An Authentication Service for Open Network Systems", USENIX Winter Conference, USENIX Association, February 1988.

に提案したKerberosの方法はそのようなシステムの一例である。Kerberosシステムでも、認証の領域を複数用意し、他の領域に行くときにはその認証サーバのチケットをもらうようにすることができる。

【０００７】可搬型コンピュータの認証には、常に認証サーバにアクセスできるとは限らないという問題があり、これに関しては下記文献に記載がある。

文献４：岩井 三剛、村田 賢一、所 真理雄：「可搬型計算機環境におけるホスト認証」、日本ソフトウェア科学会第９回大会予稿集、September 1992.

この文献に記載の方法では、通信する可能性のあるホストコンピュータとの共有鍵を暗号化したものをホストコンピュータが持ち、時々更新することで対処している。

【０００８】さらに上記文献１において、Lampson らは公開鍵暗号に基づいて認証を提案している。このLampson らの方法は公開鍵暗号に基づいた認証を用いており、必要なのは単なる公開鍵のデータベースである。データベースは分散されており、データベース自身もその公開鍵によって認証される。

【０００９】

【発明が解決しようとする課題】大規模分散コンピュータネットワークシステムの場合、そのシステム内の通信を行う全ホストコンピュータの鍵を集中管理するのは記憶容量的にもトラフィック的にも不可能であり、分散管理が必要である。また、公開鍵暗号を使うとしても、秘密鍵の安全性の問題から、各鍵データベースのための鍵は、異なるものとしなければならない。つまり、鍵データベースの鍵の転送の問題が生じ、どの鍵データベースをどのように信用して用いたかが問題となる。文献１に示したLampson らの方法は、この問題を形式的に取り扱っているが、通信を行うホストコンピュータの移動は考慮していない。つまり、最近の移動可能な通信装置の進展に応じて、携帯性にすぐれたコンピュータを用いた通信装置をそのような大規模な通信システムに接続して一時的に通信システムに組み入れたり、外したりする運用が試みられており、そのような運用に公開鍵暗号を適用する場合に、公開鍵を入手するのに多大の経路をたどることは得策ではない。しかしながら、従来の方式においては、マシン（ここでは、通信を行うコンピュータ）の移動については考慮されていない。

【００１０】また、上述した大規模分散通信システムにおける公開鍵暗号処理においては、公開鍵を入手するまでに多大の経路をたどる場合がしばしば発生し、信頼する鍵を伝達するべきマシン（ここでは、正しい情報を教えると仮定したデータベース）の数を減らすには、新たな接続を増やさなければならない。特に、地球規模の大規模なコンピュータ通信システムにおいて公開鍵鍵暗号方式を適用する場合、公開鍵を正当に入手するまでに多大の経路をたどることになり、その経路において、信頼すべきマシンが多大になることは機密性および真正性の観点から好ましくない。これを改善するため、経路を少なくするように新たな接続を増加することは設備を変更するなどの観点から好ましくなく、現実的でもない。

【００１１】

【課題を解決するための手段】本発明では、このような環境で最大限の信頼性を実現するために、新たな接続を設けずに、ホストコンピュータの移動先で直接公開鍵を交換することで、通信において信頼できると仮定しなければならないホストコンピュータの数をなるべく少なくするようにする。

【００１２】したがって、本発明によれば、公開鍵を用いて通信装置相互で暗号通信を行う公開鍵暗号処理システムであって、それぞれが自己の公開鍵と該公開鍵に対応する秘密鍵を記憶している複数の第１の通信装置と、前記複数の第１の通信装置をグループ分けし、それぞれのグループに属する前記第１の通信装置の名称とその公開鍵とを記憶し、さらに自己の公開鍵とそれに対応する秘密鍵を記憶し、そのグループに属する第１の通信装置の公開鍵の認証を行う第２の通信装置とを有し、前記第２の通信装置のそれぞれには、その通信装置と通信可能な他の第２の通信装置が登録されており、前記第１の通

信装置および前記第２の通信装置のそれぞれに、それまでに公開鍵を知った前記第１または第２の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶する手段を有し、通信装置が他の通信装置の公開鍵を入手したとき、当該記憶手段の記憶内容を更新する公開鍵暗号処理システムが提供される。好適には、上記通信装置には暗号コンピュータ通信を行うのに好適なコンピュータを含む。

【００１３】好適には、前記第１の通信装置は、自己の前記記憶手段に、その通信装置が属するグループ内の他の第１の通信装置の公開鍵を事前に登録し、かつ、登録した通信装置について信頼する通信装置が存在しないと定義し、該第１の通信装置は前記登録した第１の通信装置と、直接公開鍵を用いて通信を行う。

【００１４】また、第１の通信装置を一時的にあるグループから他のグループに変更したとき、新たなグループの第２の通信装置の前記記憶手段に、該追加した第１の通信装置の名称および公開鍵を登録する。

【００１５】または、第１の通信装置をあるグループに追加したとき、そのグループの第２の通信装置に、該追加した第１の通信装置の名称および公開鍵を登録する。

【００１６】また本発明によれば、複数の通信装置が公開鍵を用いて暗号通信する公開鍵暗号処理方法であって、前記通信装置が使用する公開鍵の認証管理をグループ分け、かつ、階層化した木構造で行い、それぞれの通信装置に自己の公開鍵とそれに対応する秘密鍵を記憶し、前記階層の上位の通信装置において、自己の公開鍵とその秘密鍵に加えて、そのグループに属する下位の通信装置の名称とその公開鍵とを記憶し、通信可能な上位の通信装置相互の関係を規定し、前記通信装置のそれぞれは、それまでに公開鍵を知った通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶、更新する公開鍵暗号処理方法が提供される。

【００１７】

【作用】信用しなければならない通信装置（コンピュータ）の数を減らすには、通信を行う通信装置の近くに通信を行おうとする通信装置を持っていき、通信相手の公開鍵を登録し、認証（識別）を行う通信装置を経由せず、直接、相手の通信装置と通信を行う。つまり、大規模通信システムにおいても、小規模通信システムと同様に、直接、公開鍵をやりとりして、認証に関与する通信装置の数を減らして、暗号通信を行う。そのため、それぞれの通信装置に上述した記憶手段を設ける。

【００１８】特に、携帯型通信装置の場合、その通信装置の属するグループが頻繁に変更になる。その変更に応じて上記記憶手段の内容を変更し、好適には、新たに属するグループ内の通信を行う相手の通信装置の公開鍵を事前に登録して、その相手と直接、暗号通信を行う。この場合、認証に関与する通信装置は介在せず、機密性か、真正性の高い通信が可能となる。

【００１９】勿論、大規模通信システムにおいても、それぞれの通信装置はどこの通信装置に対しても通信文を送ることができる。この場合、公開鍵の管理をグループ化し、階層化した木構造をとることにより、本発明による信頼性の高い公開鍵暗号処理が適用できる。

【００２０】

【実施例】本発明の公開鍵暗号処理システムについて述べる。大規模分散コンピュータ通信システムにおいては、移動するオブジェクトやホストコンピュータを含むため、ホストコンピュータの実際の位置に応じて通信相手を信頼できる度合いが変化する。信用しなければならないホストコンピュータを少なくするためには、ホストコンピュータをその場に持って行って直接通信するのが確実である。遠くのホストコンピュータを知るにはいくつかのホストコンピュータを介するしかないわけで、認証（識別）においてもそれを信用しなければならないのは当然である。つまり、確実性を求めるならば、ホストコンピュータの移動を積極的に利用して、信用するべきホストコンピュータを減らせばよい。本発明では、従来問題となっていたホストコンピュータの移動を利用して、秘密通信や認証をなるべく少ないホストコンピュータを信用して通信する。本発明は、公開鍵データベースの管理に「階層相対名前付け法の構造」を用いている。階層相対名前付け法はマシンの移動に対応した名前付け法であり、本発明の公開鍵暗号処理システムの実施例として、移動可能なマシンを含む秘密通信・認証システムを例示する。

【００２１】まず、マイグレーションがない場合（ホストコンピュータの移動がない場合）の階層鍵管理方法について述べる。認証のためには相手を特定する必要があり、それは識別コードまたは名前（ＩＤ）で表される。本発明においては、大規模分散システムに適したオブジェクトの名前付けとアドレシングの方法である、階層相対名前付け法をすでに下記文献に提案している。

文献５：藤波　順久、横手　靖彦：「大規模分散システムにおけるオブジェクトの名前づけ」、コンピュータソフトウェア、Vol. 10, No. 3 (1993), pp. 37-47.

この方法は、可搬型ホストコンピュータを含み、集中管理が不可能なほど大規模な分散コンピュータ通信システムにおいて、識別可能性、移動透過性、拡張性、高効率、可用性・耐故障性を持つオブジェクトの名前づけ法である。

【００２２】この方式では、仮定として、システムは論理的に階層構造をなしており、局所的名前空間を持っているとしている。大規模分散コンピュータ通信システム内のオブジェクトには、それが生成された名前空間（以下、「本籍」という）と現在いる名前空間（以下、「現住所」という）という概念がある。名前空間（のマネージャ）であるオブジェクトは、現在そこにいるオブジェ

クトと直接通信可能であるとする。通常、ホストコンピュータは一つの名前空間とそこにいるオブジェクトに対応しているため、ホストコンピュータの移動は名前空間の移動として手順が作られている。本発明でも、同様の仮定を用いて、階層鍵管理を行う。

【００２３】図１に示したように、各ホストコンピュータＡは自分の秘密鍵$K_A{}^{-1}$を保持しており、本籍のマネージャＭの公開鍵K_Mを知っている。また、本籍のマネージャＭはそれに属する全ホストコンピュータの公開鍵のデータベースを持ち、また、自分の秘密鍵$K_M{}^{-1}$を保持している。同じ本籍を持つホストコンピュータが通信している限り、鍵管理は単純である。すなわち、マネージャＭは識別コード（または名前）ＩＤ（Identification）で指定されたホストコンピュータの公開鍵をマネージャＭの秘密鍵でサインして返す。各ホストはマネージャＭの公開鍵を知っているので、公開鍵を取り出して相手ホストコンピュータを認証したり、秘密メッセージを送ったりできる。ホストコンピュータが異なるマネージャに属している場合には、いくつかのマネージャを順にたどっていって順番に公開鍵を得る必要がある。例えば、図１に示したホストコンピュータＡからホストコンピュータＤに秘密のメッセージを送るためにホストコンピュータＤの公開鍵を得たいとする。もしホストコンピュータＡが直接マネージャＴから公開鍵K_Dを送ってもらったとすると、これは認証されないので、間違った公開鍵が送られてきた可能性がある。

【００２４】そこで、下記手順、（１）ホストコンピュータＡはマネージャＳから鍵$K_S{}^{-1}$（K_M）を送ってもい、鍵K_Sを使って読む、（２）ホストコンピュータＡはマネージャＲから鍵$K_M{}^{-1}$（K_T）を送ってもらい、鍵K_Mを使って読む、（３）ホストコンピュータＡはマネージャＴから鍵$K_T{}^{-1}$（K_D）を送ってもらい、鍵K_Tを使って読む、を経れば、安全に公開鍵K_Dを得ることができる。

【００２５】ここで、各マネージャは「一つ向こうの」公開鍵を正しく教えると仮定している。つまり、

（ａ）ホストコンピュータＡは、「鍵K_Sで読むと、ホストコンピュータＡ以外のマネージャＳに隣接するホストコンピュータが言ったことになっている命題」を、そのホストコンピュータが本当に言ったと信じる。

（ｂ）マネージャＳは、「鍵K_Mで読むと、マネージャＳ以外のマネージャＲに隣接するホストコンピュータが言ったことになっている命題」を、そのホストコンピュータが本当に言ったと信じる。

（ｃ）マネージャＲは、「鍵K_Tで読むと、マネージャＲ以外のマネージャＴに隣接するホストコンピュータが言ったことになっている命題」を、そのホストコンピュータが本当に言ったと信じる。

ということである。これとマネージャＴのデータベースの情報である、「ホストコンピュータＤは、自分の公開鍵がK_Dであると言った」という命題を組み合わせることで、ホストコンピュータＡはホストコンピュータＤの公開鍵がK_Dであると信じるようになる。このやり方は上記文献１で形式的に述べられている。

【００２６】上記仮定が成り立たないかぎり、つまり、どこかのマネージャが嘘の公開鍵を返した場合、メッセージを相手が受け取ることができなくなってしまう。一方、マネージャは、メッセージの中継をごまかすことによってもメッセージを届かなくすることができる。つまり、マネージャが正しい公開鍵を返すかどうかは、マネージャを通るメッセージが正しく届けられるかどうかということと同程度に信用できる。換言すれば、従来の方法では、マネージャ（認証サーバ）は生成した共有鍵を用いてメッセージをこっそり盗聴することができ、これを検出するのは困難である。したがって、これは妥当な仮定である。

【００２７】次にマイグレーション（ホストコンピュータの移動）がある場合について述べる。ホストコンピュータが移動した場合、従来のやり方では、移動先でも本籍に認証してもらうか、または移動先でも認証してもらえるように本籍から手続きを行っておく必要があった。すると、ホストコンピュータの信用に関する仮定が増えてしまう。ところで、移動先で物理的接続がされるときには、そこのマネージャの公開鍵を直接入力することができる。また、同時にマネージャはホストコンピュータの公開鍵をデータベースに入れることができる。これを使うと、移動したホストコンピュータが移動先のホストコンピュータを認証する場合でも、その逆の場合でも、信用しなければならないホストコンピュータの数を減らすことができる。

【００２８】例えば図２でホストコンピュータＢが移動してマネージャＴと接続されたとする。そのときに、公開鍵K_TをホストコンピュータＢに入力し、また、マネージャＴのデータベースに鍵K_Bを登録すれば、ホストコンピュータＢは「鍵K_Tで読むと、ホストコンピュータＢ以外のマネージャＴに隣接するホストコンピュータが言ったことになっている命題」を、そのホストコンピュータが本当に言ったと信じるという仮定のもとで、そこのホストコンピュータ、例えばホストコンピュータＣの認証ができるようになる。したがって、上述した仮定（ａ）～（ｃ）は不要である。

【００２９】逆にいえば、移動先でホストコンピュータが特に公開鍵を直接入力することがなければ、本籍のマネージャから始めて今まで同じ仮定をおく必要があるし、相手がこちらを認証するには移動先のマネージャは本籍のマネージャに頼んで（やはり同様の仮定が必要）公開鍵を取り寄せなければならない。

【００３０】階層相対名前付け法におけるホストの移動手順に認証を付け加えた例を述べる。階層相対名前付け法では、本籍は常にそれに属するオブジェクトの現在位

置を正しく知っている必要があるため、切断通知、新住所通知、確認通知には認証が加わった。この例では、移動するホストコンピュータをAとする。

【0031】（1）移動開始：ホストコンピュータAは本籍のマネージャに切断通知を送る。これには、ホストコンピュータAの秘密鍵でサインしたオブジェクト識別コードＯＩＤ（Object ID ）とオブジェクトアドレスＯＡＤ（Object address）とを付け加えて認証する。このオブジェクトアドレスＯＡＤにはタイムスタンプが含まれているため、切断通知の再送は防止される。このメッセージが通過した名前空間のマネージャは、ホストAとその子孫に対する局所識別コードＬＩＤ（Local ID）とオブジェクトアドレスＯＡＤ、オブジェクト識別コードＯＩＤと局所アドレスＬＡＤ（Local Address）、または、オブジェクト識別コードＯＩＤとオブジェクトアドレスＯＡＤの組を無効にする（これは認証されなくてもよい）。

【0032】（2）移動：ホストコンピュータAが移動する。

【0033】（3）移動終了：新しい現住所のマネージャに局所アドレスＬＡＤを割り当ててもらう。マネージャはこのオブジェクト識別コードＯＩＤと局所アドレスＬＡＤの組を記憶する。このとき同時に、ホストコンピュータAは現住所のマネージャの公開鍵を記憶し、また、ホストコンピュータAの公開鍵を現住所のマネージャのデータベースに登録することが望ましい。そして、ホストコンピュータAは、本籍のマネージャにオブジェクト識別コードＯＩＤ、新しいオブジェクトアドレスＯＡＤ、仮想的なオブジェクト識別コードＯＩＤである（0：）と、それらをホストコンピュータAの秘密鍵でサインしたものを通知する。本籍のマネージャはオブジェクトアドレスＯＡＤを更新し、確認通知を返す。確認通知には、新住所通知のオブジェクトアドレスＯＡＤについていたタイムスタンプと、仮想的なオブジェクト識別コードＯＩＤ0：の現在の値（逆ＯＩＤ）が、本籍の秘密鍵でサインされたものが含まれている。

【0034】階層相対名前付け法の場合、識別コードＩＤが相対表現なので、本籍のマネージャと認証通信して相対位置を確認しないと識別コードＩＤが決められない。これはつまり、識別コードＩＤの扱いに関しては途中のホストコンピュータを信用しているということである。これは次の2つの点で問題がある。もし本籍のマネージャと通信できなかった場合どことも通信が始められないこと、および、依然として信用すべきホストコンピュータが減らないことである。このうち特に前者を解決する方法として、本籍のバックアップの働きをするホストコンピュータを用意する。後者の問題は、バックアップの識別コードＩＤの確定の問題があって解決し難いが、ホストコンピュータの移動前に移動先を決めておくことによってある程度解決できる。

【0035】オブジェクトについてマイグレーションが起きる場合は、ホストコンピュータについてと同様に、オブジェクトも認証の対象にしなければならない。ただし、移動先のホストコンピュータは信用できなければならない。ホストコンピュータはオブジェクトの全データに（秘密鍵にも）アクセスできるからである。機密情報を交信するという点では、通常の通信もオブジェクトマイグレーションも相手のホストコンピュータを同程度に信用している必要がある。ただ、外から見た場合、移動先のホストコンピュータを認証するよりもオブジェクトそのものを認証できたほうが都合がよいこと、移動先からの新住所通知をオブジェクトの鍵で認証できれば便利なことから、オブジェクトにも秘密鍵と公開鍵の組を割り当てる。

【0036】オブジェクトマイグレーションの手順は、ホストコンピュータの場合とほとんど同じである。オブジェクトが初めて本籍を離れるときに、マネージャがオブジェクトに秘密鍵を割り当て、公開鍵をデータベースに登録するという点が異なる。オブジェクトが本籍に帰るときには、公開鍵と対応する秘密鍵を破棄し、次回は別の鍵を使うこともできる。

【0037】本発明の方法では、従来の方法に比べると、移動した先にあるホストコンピュータとの通信で用いる仮定が少なくなり、より信頼できる通信が可能である。さらに、移動先のデータベースの公開鍵を覚えたままにしておけば、ホストコンピュータが元の場所に戻ってからも、この信頼性は変わらない。

【0038】以上に述べたように、本発明に基づく大規模分散システムに適した公開鍵認証のための鍵管理においては、鍵は階層管理され、鍵データベースへのアクセスの集中が防止でき、また、ホストコンピュータの移動を積極的に利用して、信用しなければならないホストコンピュータの数をできるだけ少なくした秘密・認証通信ができる。

【0039】本発明の基本技術である公開鍵鍵暗号方式について述べる。平文とは、暗号化される前のデータをいい、暗号文とは、暗号化後のデータをいう。暗号変換式Ｅκ は平文から暗号文への鍵Ｋによって決まる変換式であり、復号変換式Ｄκ' は、暗号文から平文への、鍵Ｋ' によって決まる変換式である。鍵から暗号化変換式、復号化変換式を決める手順は公開されているとする。各平文Ｍについて、

【数１】

$$D_{K'}(E_K(M))=M$$

である。

【0040】ＤＥＳ（Data Encryption Standard, FIPS PUB 46, National Bureau of Standards, Washington, D.C.（Jan. 1977））などの共有鍵暗号では、Ｋ＝Ｋ'

であるが、1976年にDiffile とHellman によって紹介された下記文献、
文献６：W. Diffie and M. Hellman, "New Direction in Cryptography ", IEEE Transactions on Infomation Theory Vol. IT-22(6) pp. 644-654 、Nov1976
に記載された公開鍵暗号方式では、鍵Ｋと鍵Ｋ'とは異なり、鍵Ｋから鍵Ｋ'を求めることは非常に困難である。
【００４１】公開鍵暗号方式は、上述したように、暗号化に使う鍵（公開鍵）と解読に使う鍵（秘密鍵）を異なるものとし公開鍵から秘密鍵を推測し難くした暗号処理方式であり、この公開鍵暗号を用いて秘密通信を行うシステムでは、各ユーザ（ホストコンピュータ）Ａは２つの鍵K_Aと鍵K_A'を持っている。鍵K_Aは公開鍵と呼ばれ、公開鍵データベースに登録されている。公開鍵データベースは、ユーザ名Ａを指定するとその公開鍵K_Aを答える機能を持つ。鍵K_A'は秘密鍵と呼ばれ、ユーザＡのみが知っている。ユーザＡが平文Ｍを秘密裡にユーザＢに送ろうとするときには、ユーザＡは公開鍵データベースから鍵K_Bを得、ユーザＢに暗号文
【数２】

$$C = E_{K_B}(M)$$

を送る。ユーザＢは、それを受信して平文
【数３】

$$M = D_{K_B'}(C)$$

を得る。秘密鍵K_B'はユーザＢのみが知っており、しかも、鍵K_Bから鍵K_B'を計算するのは非常に困難であるため、ユーザＢ以外のユーザが平文Ｍを得るのを防ぐことができる。
【００４２】公開鍵暗号を用いて認証（本人確認）を行うシステムでは、暗号化変換と復号化変換に対する仮定として、平文を復号化変換できること、暗号文を暗号化変換できること、各平文Ｍについて
【数４】

$$M = E_K(D_{K'}(M))$$

であることを要請する。上と同様に各ユーザが二つの鍵を持ち、公開鍵データベースを用意するならば、ユーザＡが平文ＭをユーザＢに、確かにユーザＡから送られたことがわかるように送るときには、ユーザＡはユーザＢに
【数５】

$$C = D_{K_A'}(M)$$

を送る。ユーザへはそれを受信して後、あるいはあらかじめ公開鍵データベースから鍵K_Aを得、それを用いて平文
【数６】

$$M = E_{K_A}(C)$$

を得る。秘密鍵K_A'はユーザＡのみが知っており、しかも、鍵K_Aから鍵K_A'を計算するのは非常に困難であるため、意味のある平文に復元できるようなコードを作れるのはユーザＡだけである。つまり、ユーザＡ以外のユーザがユーザＡのふりをして通信することはできない。
【００４３】公開鍵暗号の具体例としては、
文献７：R. L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems ", Communications ofthe ACM Vol 21(2) pp. 120-126 、 Feb. 1978
に記載されたＲＡＳ方式が知られている。
【００４４】非常に多くのコンピュータを含むネットワークシステムの場合、ユーザ数も非常に多く、公開鍵データベースをシステム内に１個だけ用意しておいた場合、データベースは巨大になり、また、データベースへのアクセス頻度が非常に高くなる。これを避けるために公開鍵データベースは分散管理するのが普通である。例えば、上述した文献１（Butler Lampson, Martin Abadi, Michael Burrows, and Edward Wobber, "Authentication in Distributed Systems: Theory andPractice ", Proceedings of the 13th ACM Symposium on Operating System Principles, October 1991）に記載されたLampson らのシステムの一例を次に示す。
【００４５】図３で、記号C_0, C_1, C_2, …C_{11}, C_{12}, …C_{21}, C_{22}, …は通信を行うコンピュータ（計算機）を示しており、鍵Ｋ，Ｋ'から暗号化変換式E_Kと復号化変換式$D_{K'}$とを求め、それを実行することができる。ここでは、ネットワークシステムに接続されているコンピュータが３段の木構造に管理されているとする。つまり、記号C_1はコンピュータC_1, C_{11}, C_{12}, …を代表するコンピュータを示しており、コンピュータC_{11}, C_{12}, …の公開鍵K_{11}, K_{12}, …を記憶している公開鍵データベースＤB_1を持っている。同様に、コンピュータC_2, はコンピュータC_2, C_{21} C_{22}, …を代表するコンピュータであり、コンピュータC_{21}, C_{22}, …の公開鍵K_{21}, K_{22}, …を記憶している公開鍵データベースＤB_2を持っている。また、コンピュータC_0 はコンピュータC_0 , C_1, C_2, …を代表するコンピュータであり、コンピュータC_1, C_2, …の公開鍵K_1, K_2, …を記憶している公開鍵データベースＤB_0を持っている。各コンピュータは一人のユーザが使用しているとする。つまり、各コンピュータC_nは公開鍵K_nと秘密鍵K_n'を記憶している。公開鍵を公開鍵データベースに登録する作業は、人手で本人確認をして行う。

【００４６】このシステムでは、公開鍵データベースから公開鍵を得るときにもネットワークを使用する。従って、公開鍵データベースを持つコンピュータ以外から偽の公開鍵を与えられることを防ぐために、公開鍵データベースも認証の対象にする。例えば、コンピュータC_{11}には、自分の公開鍵と秘密鍵のほかに公開鍵データベースDB_1を持つコンピュータC_1の公開鍵K_1が予め入手で入力されている。コンピュータC_{12}の公開鍵を得たい場合、「コンピュータC_{12}の公開鍵はK_{12}である。」という通信文をM_{12}とすれば、コンピュータC_1からコンピュータC_{11}に復号文$D_{K1}{}'(M_{12})$を送ってもらえばよい。コンピュータC_{11}は公開鍵K_1を用いて通信文$M_{12}=E_{K1}(D_{K1}{}'(M_{12}))$を得、確かにコンピュータ$C_1$から送られたものと確認することができる。ここでは、コンピュータC_1は十分に信頼でき、コンピュータC_{11}, C_{12}, …の公開鍵K_{11}, K_{12}, …を正しく教えると仮定している。

【００４７】代表となるコンピュータが異なるようなコンピュータの公開鍵を得ようとする場合は、いくつかの公開鍵データベースを順に検索していく必要がある。例えば、コンピュータC_{11}がコンピュータC_{21}の公開鍵を得たい場合、「コンピュータC_0の公開鍵はK_0である。」という通信文をM_0とすれば、まずコンピュータC_1に頼んで$D_{K1}{}'(M_0)$を送ってもらう。コンピュータC_{11}は公開鍵K_1を用いて$M_0=E_{K1}(D_{K1}(M_0))$を得る。次に、「コンピュータC_2の公開鍵はK_2である。」という通信文をM_2とするとき、コンピュータC_0に頼んで$D_{K0}(M_2)$を送ってもらう。コンピュータC_{11}は先ほど得た公開鍵K_0を用いて$M_2=E_{K0}(D_{K0}(M_2))$を得る。最後に、「コンピュータC_{21}の公開鍵はK_{21}である。」という通信文をM_{21}とするとき、コンピュータC_2に頼んで$D_{K2}{}'(M_{21})$を送ってもらう。コンピュータC_{11}は先ほど得た公開鍵K_2を用いて$M_{21}=E_{K2}(D_{K2}{}'(M_{21}))$を得る。こうしてコンピュータ$C_{21}$の公開鍵$K_{21}$を得ることができる。ここでは、コンピュータ$C_1$, C_0, C_2, は十分に信頼でき、その一つ向こうのコンピュータの公開鍵を正しく教えると仮定している。

【００４８】信頼するべき経路を辿るコンピュータの数を減らしたい場合には、公開鍵をあらかじめ知らせてあるコンピュータを、木構造で管理されている関係より増やせばよい。例えば、コンピュータC_2の公開鍵K_2をあらかじめ人手でコンピュータC_1に入力しておけば、コンピュータC_0を信頼するという仮定は必要なくなる。ただし、コンピュータC_1がコンピュータC_2の公開鍵を知っているという情報は、何らかの形でコンピュータC_{11}が知る必要がある。

【００４９】文献１に示したLampson らのシステムの問題点は、上述したように移動可能なコンピュータについて考慮されていないことである。例えば、コンピュータC_{11}が移動可能な計算機であって、コンピュータC_{11}のユーザがコンピュータC_{11}をコンピュータC_{21}の近くに持って行ったとする。このときもコンピュータC_{11}からコンピュータC_{21}に秘密の通信文を送ろうとするときには、コンピュータC_1, C_0, C_2 と順に通信してコンピュータC_{21}の公開鍵K_{21}を得なければならないし、コンピュータC_1, C_0, C_2 は信頼できると仮定しなければならない。逆に、コンピュータC_{21}からコンピュータC_{11}に秘密の通信文を送ろうとするときにも、コンピュータC_2, C_0, C_1 と順に通信してコンピュータC_{11}の公開鍵K_{11}を得なければならないし、コンピュータC_2, C_0, C_1 は信頼できると仮定しなければならない。

【００５０】本発明では、上の問題を解決するために、各コンピュータはそれまでに公開鍵を知ったコンピュータの名称とその公開鍵、そしてそれを知るために信頼したコンピュータの名称の対応表（テーブル）を持っているとする。対応表は公開鍵データベースから公開鍵を教えてもらったときのほか、コンピュータのユーザがコンピュータの名称とその公開鍵を入力したとき（このときには信頼したコンピュータの名称は空である）にも更新される。特に、移動可能なコンピュータの場合には、他のコンピュータの近くに移動したときに、そのコンピュータのユーザから直接公開鍵を教えてもらえることが期待できるので、信頼したコンピュータの数の少ない公開鍵を得ることができる。

【００５１】例えば、図４でコンピュータC_{11}のユーザがコンピュータC_{11}をコンピュータC_{21}の近くに持って行ったときに、コンピュータC_{21}のユーザからその公開鍵K_{21}を教えてもらったとする。すると、他のコンピュータを信頼することなく、コンピュータC_{11}からコンピュータC_{21}に秘密の通信文を送ることができる。この状況は、コンピュータC_{11}を元の場所に持って帰ったり、さらに他の場所に移動したりしてからでも変わらない。また逆に、コンピュータC_{11}のユーザがコンピュータC_{21}のユーザに公開鍵K_{11}を教えれば、他のコンピュータを信頼することなく、C_{21}からC_{11}に秘密の通信文を送ることができるし、コンピュータC_{11}をまた移動してからでも同様である。

【００５２】他の例として、コンピュータC_{11}をコンピュータC_2の近くに持って行き、公開鍵K_2を入力したとする。すると、コンピュータC_{22}に秘密の通信文を送りたい場合、すでに対応表に、「コンピュータC_2の公開鍵はK_2で、信頼したコンピュータはない」という情報があるため、コンピュータC_2と通信して公開鍵K_{22}を教えてもらうだけでよい。この場合、信頼したコンピュータはコンピュータC_2のみである。このようにして、公開鍵を直接入力したコンピュータを中心としたいくつかのコンピュータについては、その公開鍵を得るために信頼したコンピュータの数を、Lampson らのシステムに比べて少なくすることができる。

【００５３】公開鍵を得たいコンピュータの名前と、すでに対応表に記録されているコンピュータの名前から、どのコンピュータと順に通信していくかを決定するために、コンピュータの名前付けはコンピュータの管理の木構造、つまり、鍵管理の木構造を反映したものでなければならない。これは、Lampson らのシステムでも同様である。

【００５４】図４で、コンピュータC_0, C_1, C_2,…は据え置き型コンピュータであり、C_{11}, C_{12}, …C_{21}, C_{22}, …は据え置き型、あるいは移動可能なコンピュータである。各コンピュータは、鍵K，K'から暗号化変換式E_kと復号化変換式D_k'を求め、それを実行することができる。これらのコンピュータはコンピュータネットワークシステムに含まれており、各コンピュータは、システム内の他のコンピュータに通信文を送ることができる。コンピュータが移動しても通信文は相手のコンピュータに到着するものとする。ネットワークシステムの各通信路では、通信文の送信者と受信者以外の者に通信内容を読み取られたり、通信内容を改変されたりする可能性がある。

【００５５】コンピュータネットワークシステムに接続されているコンピュータは、鍵管理の３段の木構造に管理されているとする。つまり、コンピュータC_1はコンピュータC_{11}, C_{12}, …を代表するコンピュータであり、コンピュータC_{11}, C_{12}, …の公開鍵K_{11}, K_{12}, …を記憶している公開鍵データベースDB_1を持っている。同様に、コンピュータC_2はコンピュータC_{21}, C_{22}, …を代表するコンピュータであり、コンピュータC_{21}, C_{22}, …の公開鍵K_{21}, K_{22}, を記憶している公開鍵データベースDB_2を持っている。また、コンピュータC_0はコンピュータC_1, C_2 …を代表するコンピュータであり、コンピュータC_1, C_2…の公開鍵K_1, K_2,…を記憶している公開鍵データベースDB_0を持っている。各コンピュータは一人のユーザが使用しているとする。つまり、各コンピュータC_nは公開鍵K_nと秘密鍵K_n'を記憶している。公開鍵を公開鍵データベースに登録する作業は、人手で本人確認をして行う。

【００５６】各データベースが管理しているコンピュータと鍵の数はたとえば、１０００～１００００程度とする。公開鍵暗号としてＲＳＡ方式を用いた場合、公開鍵は２００桁の１０進数２個（１．３キロビット）程度の記憶領域を必要とするため、データベースの大きさは１．３～１３メガビット程度となる。

【００５７】各コンピュータC_0, C_1, C_2, (一般的にはC_nとして表す)は、表（テーブル）T_0, T_1, T_2, (一般的にはT_nとして表す)を持つ。対応表の各行には、コンピュータの名前（名称）、公開鍵、使われた時刻、信頼したコンピュータの名称（集合）が記録されている。対応表の行数は１００～１０００程度とし、それを越える行を記憶させようとした場合には、信頼したコンピュータの名称の数が多い順、数が同じものが複数あるときには時刻の古い順に消去する。つまり、新しいものを残しておく。コンピュータの名称の大きさなどを無視すれば、対応表の大きさは１３０キロビット～１．３メガビット程度となる。

【００５８】コンピュータC_nのユーザがコンピュータC_pの公開鍵K_pを入力したときの表更新手順を図５に示す。まず、対応表T_nにコンピュータC_pについての行があるとき（ステップ１）は、コンピュータC_pについての行を作成する（ステップ２）。ないときには、その行の公開鍵とK_pを比べる（ステップ３）。異なる場合には、信頼したコンピュータのうちのどれかが嘘をついたことがわかるため、その旨エラー表示した後（ステップ４）、信頼したコンピュータの名称にコンピュータC_pを含む、対応表T_nのすべての行を削除する（ステップ５）。一致したなら、その行の信頼したコンピュータの名称をC, C' C''…とするとき（ステップ６），対応表T_nの各行のうち、信頼したコンピュータの名称にC_p, C, C', C''…が含まれるなら、コンピュータC, C', C''…を除く（ステップ７）。最後にコンピュータC_pについての行の公開鍵にK_pを、使われた時刻に現在時刻を、信頼した計算機名に空集合を設定する（ステップ８）。

【００５９】コンピュータC_nのユーザがC_pの公開鍵を得ようとしたときの表更新手順を図６に示す。まず、コンピュータC_pがコンピュータC_nの代表となるコンピュータであった場合は、すでに公開鍵を知っているので終了する（ステップ１１）。対応表T_nにコンピュータC_pについての行があるとき（ステップ１２）はそれを使えばよいので終了する。ないときには、対応表T_nの各行と代表となるコンピュータについて、コンピュータC_pから木構造をたどったときの段数と信頼したコンピュータの数（代表となるコンピュータの場合は０）の和を計算し（ステップ１３）、和の最も少ないコンピュータをC_qとする（ステップ１４）。コンピュータの数が複数あるときは段数の少ないものから任意に一つを選ぶ。コンピュータC_qに木構造で隣接し、コンピュータC_pに一段近いコンピュータ（一意に決まる）をコンピュータC_rとする（ステップ１５）。対応表T_nのコンピュータC_qについての行の使われた時刻を現在時刻とする（ステップ１６）。コンピュータC_qと通信してコンピュータC_rの公開鍵K_rを得る（ステップ１７）。コンピュータC_qについての行の信頼したコンピュータの名称をC, C', C''…とする（ステップ１８）。対応表T_nにコンピュータC_rについての行を作成し（ステップ１９）、公開鍵にK_rを、使われた時刻に現在時刻を、信頼したコンピュータの名称にコンピュータC, C', C''…を設定する（ステップ２０）。そしてステップ１２に戻る。

【００６０】以上のように処理することにより、ホスト

コンピュータの移動、あるいは、オブジェクトの移動があっても、大規模分散コンピュータ通信ネットワークシステムにおいて、信頼するコンピュータの数を減らして信頼性の高い通信を行うことができる。

【００６１】

【発明の効果】本発明によれば、ユーザが望むならば直接公開鍵をやりとりすることで、大規模分散システムでも、小規模なシステムと同様の信頼性を得ることができる。また、それほど信頼性を要求しない場合は、間接的に公開鍵を得ることもでき、ユーザの要求レベルに合わせた信頼性で秘密通信・認証が可能になる。

【００６２】また本発明に基づく大規模分散システムに適した公開鍵認証のための鍵管理においては、鍵は階層管理され、鍵データベースへのアクセスの集中が防止でき、また、ホストコンピュータの移動を積極的に利用して、信用しなければならないホストコンピュータの数をできるだけ少なくした秘密・認証通信ができる。

【００６３】さらに本発明の方式は、従来の方式に比べると、移動した先にあるホストコンピュータとの通信で用いる仮定が少なくなり、より信頼できる通信が可能である。さらに、移動先のデータベースの公開鍵を覚えたままにしておけば、ホストコンピュータが元の場所に戻ってからも、この信頼性は変わらない。

【図面の簡単な説明】

【図１】従来の公開鍵暗号処理システムの構成図である。

【図２】本発明の公開鍵暗号処理システムの構成図である。

【図３】従来の公開鍵暗号処理システムの構成図である。

【図４】本発明の公開鍵暗号処理システムの構成図である。

【図５】図４に示した公開鍵暗号処理システムにおける公開鍵を直接入力する場合の処理方法を示すフローチャートである。

【図６】図４に示した公開鍵暗号処理システムにおける公開鍵を入手するときの更新手順を示すフローチャートである。

【符号の説明】

C_0，C_1，C_2，C_m・・通信を行うコンピュータ

T_0，T_1，T_2，T_n・・通信を行うコンピュータ内の対応表

DB_0，DB_1，DB_2，DB_n・・データベース

K_0，K_1，K_2，K_m・・鍵

【公報種別】特許法第１７条の２の規定による補正の掲載
【部門区分】第７部門第３区分
【発行日】平成１３年７月６日（２００１．７．６）

【公開番号】特開平７－３８５５５
【公開日】平成７年２月７日（１９９５．２．７）
【年通号数】公開特許公報７－３８６
【出願番号】特願平５－１５５５７９
【国際特許分類第７版】
　H04L　9/00
　G09C　1/00
【ＦＩ】
　H04L　9/00
　G09C　1/00

【手続補正書】
【提出日】平成１２年６月２６日（２０００．６．２６）
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】全文
【補正方法】変更
【補正内容】
【書類名】　明細書
【発明の名称】　通信装置と方法、通信管理装置と方法、並びに公開鍵暗号処理システムと方法
【特許請求の範囲】
【請求項１】公開鍵を用いて暗号通信を行う通信装置であって、
自己の公開鍵と該公開鍵に対応する秘密鍵を記憶し、それまでに公開鍵を知った他の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶する手段を有し、
他の通信装置の公開鍵を入手したとき、当該記憶手段の記憶内容を更新する
通信装置。
【請求項２】前記記憶手段に、属するグループ内の他の通信装置の公開鍵を事前に登録し、かつ、登録した通信装置について信頼する通信装置が存在しないと定義し、前記登録した通信装置と、直接公開鍵を用いて通信を行う
請求項１記載の通信装置。
【請求項３】あるグループに属し、該当するグループを管理する上位装置に名称とその公開鍵が管理され、公開鍵の認証が前記上位装置により行われ、
属するグループを他のグループに一時的に変更したとき、
新たなグループの上位装置に名称および公開鍵を登録する
請求項１または２記載の通信装置。
【請求項４】あるグループに追加したとき、そのグループの上位装置に、名称および公開鍵を登録する
請求項１または２記載の通信装置。
【請求項５】公開鍵を用いて暗号通信を行う通信方法であって、
自己の公開鍵と該公開鍵に対応する秘密鍵を記憶し、
それまでに公開鍵を知った他の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶し、
他の通信装置の公開鍵を入手したとき、記憶内容を更新する
通信方法。
【請求項６】自己の公開鍵と該公開鍵に対応する秘密鍵を記憶し、公開鍵を用いて暗号通信を行う通信装置が使用する公開鍵の認証管理をグループ分け、かつ、階層化した木構造で行う通信管理装置であって、
前記認証管理する通信装置の名称とその公開鍵とを記憶し、さらに自己の公開鍵とそれに対応する秘密鍵を記憶し、それまでに公開鍵を知った前記第１または第２の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶する手段を有し、
管理する通信装置と通信可能な他の通信管理装置が登録されており、
他の装置の公開鍵を入手したとき、当該記憶手段の記憶内容を更新する
通信管理装置。
【請求項７】管理するグループに他のグループに属する通信装置が一時的に変更されとき、
変更された通信装置の名称および公開鍵を登録する
請求項６記載の通信管理装置。
【請求項８】管理するグループに通信装置が追加されたとき、その通信装置の名称および公開鍵を登録する
請求項６記載の通信装置。
【請求項９】自己の公開鍵と該公開鍵に対応する秘密鍵

を記憶し、公開鍵を用いて暗号通信を行う通信装置が使用する公開鍵の認証管理をグループ分け、かつ、階層化した木構造で行う通信管理方法であって、
前記認証管理する通信装置の名称とその公開鍵とを記憶し、さらに自己の公開鍵とそれに対応する秘密鍵を記憶し、
それまでに公開鍵を知った前記第１または第２の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶し、
管理する通信装置と通信可能な他の通信管理装置を登録し、
他の装置の公開鍵を入手したとき、記憶内容を更新する
通信管理方法。

【請求項１０】公開鍵を用いて通信装置相互で暗号通信を行う公開鍵暗号処理システムであって、
それぞれが自己の公開鍵と該公開鍵に対応する秘密鍵を記憶している複数の第１の通信装置と、
前記複数の第１の通信装置をグループ分けし、それぞれのグループに属する前記第１の通信装置の名称とその公開鍵とを記憶し、さらに自己の公開鍵とそれに対応する秘密鍵を記憶し、そのグループに属する第１の通信装置の公開鍵の認証を行う第２の通信装置と
を有し、
前記第２の通信装置のそれぞれには、その通信装置と通信可能な他の第２の通信装置が登録されており、
前記第１の通信装置および前記第２の通信装置のそれぞれに、それまでに公開鍵を知った前記第１または第２の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶する手段を有し、
通信装置が他の通信装置の公開鍵を入手したとき、当該記憶手段の記憶内容を更新する
公開鍵暗号処理システム。

【請求項１１】前記第１の通信装置は、自己の前記記憶手段に、その通信装置が属するグループ内の他の第１の通信装置の公開鍵を事前に登録し、かつ、登録した通信装置について信頼する通信装置が存在しないと定義し、該第１の通信装置は前記登録した第１の通信装置と、直接公開鍵を用いて通信を行う
請求項１０記載の公開鍵暗号処理システム。

【請求項１２】第１の通信装置をあるグループから他のグループに一時的に変更したとき、
新たなグループの第２の通信装置の前記記憶手段に、該追加した第１の通信装置の名称および公開鍵を登録する
請求項１０または１１記載の公開鍵暗号処理システム。

【請求項１３】第１の通信装置をあるグループに追加したとき、そのグループの第２の通信装置に、該追加した第１の通信装置の名称および公開鍵を登録する
請求項１０または１１記載の公開鍵暗号処理システム。

【請求項１４】複数の通信装置が公開鍵を用いて暗号通信する公開鍵暗号処理方法であって、前記通信装置が使用する公開鍵の認証管理をグループ分け、かつ、階層化した木構造で行い、
それぞれの通信装置に自己の公開鍵とそれに対応する秘密鍵を記憶し、
前記階層の上位の通信装置において、自己の公開鍵とその秘密鍵に加えて、そのグループに属する下位の通信装置の名称とその公開鍵とを記憶し、
通信可能な上位の通信装置相互の関係を規定し、
前記通信装置のそれぞれは、それまでに公開鍵を知った通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶、更新する
公開鍵暗号処理方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は公開鍵暗号（Public-key encryption ）処理システムに関するものである。特に、本発明は公開鍵暗号を用いて秘密通信や認証を行う大規模分散コンピュータ通信システムにおいて、公開鍵のデータベースを分散管理する際に、通信先の公開鍵を知っているデータベースと直接公開鍵を交換することによってシステム内で信頼するべきマシン（コンピュータ）の交換数を減らすことが可能な公開鍵暗号処理システムに関する。

【０００２】

【従来の技術】たとえば、非常に多数、例えば１億台ものコンピュータを含むような世界規模のコンピュータネットワークシステムにおいて、各コンピュータは、システム内の他のコンピュータに通信文を送ることが可能なようにすることが試みられている。このような大規模コンピュータネットワークシステムにおいては、コンピュータとしては据え置き型のほか、移動可能なものも含まれており、コンピュータが移動しても通信文が相手のコンピュータに正確に到着するようにされている必要がある。このようなコンピュータネットワークシステムの各通信路では、通信文の送信者と受信者以外の者に通信内容を読み取られたり、通信内容を改変されたりする可能性がある。このような大規模分散コンピュータシステムは移動するオブジェクトやホストを含むため、実際の位置に応じて通信相手を信頼できる程度が変化する。

【０００３】機密情報を通信するシステムにおいては、通信の秘密性(secrecy）と真正性(authenticity)を保証することが不可欠である。これまで通信の秘密性と真正性を保証する種々の暗号化方法と復号方法とが提案されている。たとえば、公開鍵暗号方式は、暗号化に使う鍵（公開鍵）と解読に使う鍵（秘密鍵）を異なるものとし、公開鍵から秘密鍵を推測し難くした暗号処理方式である。公開鍵暗号方式においては、公開鍵暗号方式を適用するそれぞれの各ユーザは固有の公開鍵と秘密鍵を持ち、二人のユーザは互いに相手の公開鍵を知るだけで秘

密通信ができる。また逆に、秘密鍵を用いて変換した暗号文は、対応する公開鍵で読むことができる、そのユーザだけの作れる通信文となるため、ユーザの認証（本人確認）に使うこともできる。従って、公開鍵暗号を使って秘密通信や認証を行うには、ユーザの公開鍵の登録された信頼できるデータベースがあればよい。そのため、データベース用の公開鍵を各ユーザにあらかじめ知らせておき、データベースからの情報は対応する秘密鍵で変換して送るようにする。

【０００４】大規模コンピュータネットワーク通信システムの場合、データベースの規模やアクセスの集中などの問題から、分散管理する必要がある。公開鍵を分散管理するシステムの例が、下記の文献に述べられている。

文献１：Butler Lampson, Martin Abadi, Michael Burrows, and Edward Wobber, "Authentication in Distributed Systems: Theory and Practice", Proceedings of the 13th ACM Symposium on Operating System Principles, October 1991.

この文献に記載されているシステムでは、データベースは木構造状につながっており、公開鍵のデータベース自身の公開鍵は隣接するデータベースに含まれている。通信しようとする者が直接交信できるデータベースに通信相手の公開鍵がない場合、隣接するデータベースの公開鍵を順に得ていくことを、通信相手の公開鍵が含まれているデータベースに達するまで続ける。この場合、各データベースは一つ向こうのデータベースの公開鍵を正しく教えると仮定している。仮定を緩らしたい場合、木構造以外の接続（cross link）数を増やす。

【０００５】また、共有鍵暗号をもとにした方法が下記文献に提案されている。

文献２：R. M. Needham and M. D. Schroeder: "Using Encryption for Authentication in Large Network of Computers", Communications of the ACM, Vol.21, No.12(1978), pp.993-999.

このNeedham らの方法では、一つ、あるいは相互に信頼し合える認証サーバに各ホストとの共有鍵のデータベースを置く。そして、認証サーバは、要求があるごとにホストの対の間の通信のための共有鍵（セッションキー）を発行する。

【０００６】さらに、

文献３：Jennifer G. Steiner, Clifford Neuman, and Jeffery I. Schiller: Kerberos: "An Authentication Service for Open Network Systems", USENIX Winter Conference, USENIX Association, February 1988.

に提案したKerberosの方法はそのようなシステムの一例である。Kerberosシステムでも、認証の領域を複数用意し、他の領域に行くときにはそこの認証サーバのチケットをもらうようにすることができる。

【０００７】可撤型コンピュータの認証には、常に認証サーバにアクセスできるとは限らないという問題があり、これに関しては下記文献に記載がある。

文献４：岩井 三剛、村田 賢一、所 真理雄：「可搬型計算機環境におけるホスト認証」、日本ソフトウェア科学会第９回大会予稿集、September 1992.

この文献に記載の方法では、通信する可能性のあるホストコンピュータとの共有鍵を暗号化したものをホストコンピュータが持ち、時々更新することで対処している。

【０００８】さらに上記文献１において、Lampson らは公開鍵暗号に基づいて認証を提案している。このLampson らの方法は公開鍵暗号に基づいた認証を用いており、必要なのは単なる公開鍵のデータベースである。データベースは分散されており、データベース自身もその公開鍵によって認証される。

【０００９】

【発明が解決しようとする課題】大規模分散コンピュータネットワークシステムの場合、そのシステム内の通信を行う全ホストコンピュータの鍵を集中管理するのは記憶容量的にもトラフィック的にも不可能であり、分散管理が必要である。また、公開鍵暗号を使うとしても、秘密鍵の安全性の問題から、各鍵データベースのための鍵は、異なるものとしなければならない。つまり、鍵データベースの鍵の配送の問題が生じ、どの鍵データベースをどのように信用して用いたかが問題となる。文献１に示したLampson らの方法は、この問題を形式的に取り扱っているが、通信を行うホストコンピュータの移動は考慮していない。つまり、最近の移動可能な通信装置の進展に応じて、携帯性にすぐれたコンピュータを用いた通信装置をそのような大規模な通信システムに接続して一時的に通信システムに組み入れたり、外したりする運用が試みられており、そのような運用に公開鍵暗号を適用する場合に、公開鍵を入手するのに多大の経路を辿ることは得策ではない。しかしながら、従来の方式においては、マシン（ここでは、通信を行うコンピュータ）の移動については考慮されていない。

【００１０】また、上述した大規模分散通信システムにおける公開鍵暗号処理においては、公開鍵を入手するまでに多大の経路をたどる場合がしばしば発生し、信頼する鍵を伝達するべきマシン（ここでは、正しい情報を教えると仮定したデータベース）の数を減らすには、新たな接続を増やさなければならない。特に、地球規模の大規模なコンピュータ通信システムにおいて公開鍵暗号方式を適用する場合、公開鍵を正当に入手するまでに多大の経路をたどることになり、その経路において、信頼すべきマシンが多大になることは機密性および真正性の観点から好ましくない。これを改善するため、経路を少なくするように新たな接続を増加することは設備を変更するなどの観点から好ましくなく、現実的でもない。

【００１１】

【課題を解決するための手段】本発明では、このような環境で最大限の信頼性を実現するために、新たな接続を

設けずに、ホストコンピュータの移動先で直接公開鍵を交換することで、通信において信頼できると仮定しなければならないホストコンピュータの数をなるべく少なくするようにする。

【００１２】したがって、本発明によれば、公開鍵を用いて暗号通信を行う通信装置であって、自己の公開鍵と該公開鍵に対応する秘密鍵を記憶し、それまでに公開鍵を知った他の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶する手段を有し、他の通信装置の公開鍵を入手したとき、当該記憶手段の記憶内容を更新する通信装置が提供される。

【００１３】好適には、前記記憶手段に、属するグループ内の他の通信装置の公開鍵を事前に登録し、かつ、登録した通信装置について信頼する通信装置が存在しないと定義し、前記登録した通信装置と、直接公開鍵を用いて通信を行う。

【００１４】また、あるグループに属し、該当するグループを管理する上位装置に名称とその公開鍵が管理され、公開鍵の認証が前記上位装置により行われ、属するグループを他のグループに一時的に変更したとき、新たなグループの上位装置に名称および公開鍵を登録する。

【００１５】または、あるグループに追加したとき、そのグループの上位装置に、名称および公開鍵を登録する。

【００１６】また、本発明によれば、公開鍵を用いて暗号通信を行う通信方法であって、自己の公開鍵と該公開鍵に対応する秘密鍵を記憶し、それまでに公開鍵を知った他の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶し、他の通信装置の公開鍵を入手したとき、記憶内容を更新する通信方法が提供される。

【００１７】また、本発明によれば、自己の公開鍵と該公開鍵に対応する秘密鍵を記憶し、公開鍵を用いて暗号通信を行う通信装置が使用する公開鍵の認証管理をグループ分け、かつ、階層化した木構造で行う通信管理装置であって、前記認証管理する通信装置の名称とその公開鍵とを記憶し、さらに自己の公開鍵とそれに対応する秘密鍵を記憶し、それまでに公開鍵を知った前記第１または第２の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶する手段を有し、管理する通信装置と通信可能な他の通信管理装置が登録されており、他の装置の公開鍵を入手したとき、当該記憶手段の記憶内容を更新する通信管理装置が提供される。

【００１８】また、管理するグループに他のグループに属する通信装置が一時的に変更されとき、変更された通信装置の名称および公開鍵を登録する。

【００１９】または、管理するグループに通信装置が追加されたとき、その通信装置の名称および公開鍵を登録する。

【００２０】また、本発明によれば、自己の公開鍵と該公開鍵に対応する秘密鍵を記憶し、公開鍵を用いて暗号通信を行う通信装置が使用する公開鍵の認証管理をグループ分け、かつ、階層化した木構造で行う通信管理方法であって、前記認証管理する通信装置の名称とその公開鍵とを記憶し、さらに自己の公開鍵とそれに対応する秘密鍵を記憶し、それまでに公開鍵を知った前記第１または第２の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶し、管理する通信装置と通信可能な他の通信管理装置を登録し、他の装置の公開鍵を入手したとき、記憶内容を更新する通信管理方法が提供される。

【００２１】また、本発明によれば、公開鍵を用いて通信装置相互で暗号通信を行う公開鍵暗号処理システムであって、それぞれが自己の公開鍵と該公開鍵に対応する秘密鍵を記憶している複数の第１の通信装置と、前記複数の第１の通信装置をグループ分けし、それぞれのグループに属する前記第１の通信装置の名称とその公開鍵とを記憶し、さらに自己の公開鍵とそれに対応する秘密鍵を記憶し、そのグループに属する第１の通信装置の公開鍵の認証を行う第２の通信装置とを有し、前記第２の通信装置のそれぞれには、その通信装置と通信可能な他の第２の通信装置が登録されており、前記第１の通信装置および前記第２の通信装置のそれぞれに、それまでに公開鍵を知った前記第１または第２の通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶する手段を有し、通信装置が他の通信装置の公開鍵を入手したとき、当該記憶手段の記憶内容を更新する公開鍵暗号処理システムが提供される。好適には、上記通信装置には暗号コンピュータ通信を行うのに好適なコンピュータを含む。

【００２２】好適には、前記第１の通信装置は、自己の前記記憶手段に、その通信装置が属するグループ内の他の第１の通信装置の公開鍵を事前に登録し、かつ、登録した通信装置について信頼する通信装置が存在しないと定義し、該第１の通信装置は前記登録した第１の通信装置と、直接公開鍵を用いて通信を行う。

【００２３】また、第１の通信装置を一時的にあるグループから他のグループに変更したとき、新たなグループの第２の通信装置の前記記憶手段に、該追加した第１の通信装置の名称および公開鍵を登録する。

【００２４】または、第１の通信装置をあるグループに追加したとき、そのグループの第２の通信装置に、該追加した第１の通信装置の名称および公開鍵を登録する。

【００２５】また本発明によれば、複数の通信装置が公開鍵を用いて暗号通信する公開鍵暗号処理方法であって、前記通信装置が使用する公開鍵の認証管理をグループ分け、かつ、階層化した木構造で行い、それぞれの通信装置に自己の公開鍵とそれに対応する秘密鍵を記憶

し、前記階層の上位の通信装置において、自己の公開鍵とその秘密鍵に加えて、そのグループに属する下位の通信装置の名称とその公開鍵とを記憶し、通信可能な上位の通信装置相互の関係を規定し、前記通信装置のそれぞれは、それまでに公開鍵を知った通信装置の名称とその公開鍵、および、その公開鍵を知るために信頼した通信装置の名称を記憶、更新する公開鍵暗号処理方法が提供される。

【００２６】

【作用】信用しなければならない通信装置（コンピュータ）の数を減らすには、通信を行う通信装置の近くに通信を行おうとする通信装置を持っていき、通信相手の公開鍵を登録し、認証（識別）を行う通信装置を経由せず、直接、相手の通信装置と通信を行う。つまり、大規模通信システムにおいても、小規模通信システムと同様に、直接、公開鍵をやりとりして、認証に関与する通信装置の数を減らして、暗号通信を行う。そのため、それぞれの通信装置に上述した記憶手段を設ける。

【００２７】特に、携帯型通信装置の場合、その通信装置の属するグループが頻繁に変更になる。その変更に応じて上記記憶手段の内容を変更し、好適には、新たに属するグループ内の通信を行う相手の通信装置の公開鍵を事前に登録して、その相手と直接、暗号通信を行う。この場合、認証に関与する通信装置は介在せず、機密性か、真正性の高い通信が可能となる。

【００２８】勿論、大規模通信システムにおいても、それぞれの通信装置はどこの通信装置に対しても通信文を送ることができる。この場合、公開鍵の管理をグループ化し、階層化した木構造をとることにより、本発明による信頼性の高い公開鍵暗号処理が適用できる。

【００２９】

【実施例】本発明の公開鍵暗号処理システムについて述べる。大規模分散コンピュータ通信システムにおいては、移動するオブジェクトやホストコンピュータを含むため、ホストコンピュータの実際の位置に応じて通信相手を信頼できる度合いが変化する。信用しなければならないホストコンピュータを少なくするためには、ホストコンピュータをその場に持って行って直接通信するのが確実である。遠くのホストコンピュータを知るにはいくつかのホストコンピュータを介するしかないわけで、認証（識別）においてもそれを信用しなければならないのは当然である。つまり、確実性を求めるならば、ホストコンピュータの移動を積極的に利用して、信用するべきホストコンピュータを減らせばよい。本発明では、従来問題となっていたホストコンピュータの移動を利用して、秘密通信や認証をなるべく少ないホストコンピュータを信用して通信する。本発明は、公開鍵データベースの管理に「階層相対名前付け法の構造」を用いている。階層相対名前付け法はマシンの移動に対応した名前付け法であり、本発明の公開鍵暗号処理システムの実施例として、移動可能なマシンを含む秘密通信・認証システムを例示する。

【００３０】まず、マイグレーションがない場合（ホストコンピュータの移動がない場合）の階層鍵管理方法について述べる。認証のためには相手を特定する必要があり、それは識別コードまたは名前（ＩＤ）で表される。本発明においては、大規模分散システムに適したオブジェクトの名前付けとアドレシングの方法である、階層相対名前付け法をすでに下記文献に提案している。

文献５：藤波　順久、横手　靖彦：「大規模分散システムにおけるオブジェクトの名前づけ」、コンピュータソフトウェア、Vol.10, No.3(1993), pp.37-47.

この方法は、可搬型ホストコンピュータを含み、集中管理が不可能なほど大規模な分散コンピュータ通信システムにおいて、識別可能性、移動透過性、拡張性、高効率、可用性・耐故障性を持つオブジェクトの名前づけ法である。

【００３１】この方式では、仮定として、システムは論理的に階層構造をなしており、局所的名前空間を持っているとしている。大規模分散コンピュータ通信システム内のオブジェクトには、それが生成された名前空間（以下、「本籍」という）と現在いる名前空間（以下、「現住所」という）という概念がある。名前空間（のマネージャ）であるオブジェクトは、現在そこにいるオブジェクトと直接通信可能であるとする。通常、ホストコンピュータは一つの名前空間とそこにいるオブジェクトに対応しているため、ホストコンピュータの移動は名前空間の移動として手順が作られている。本発明でも、同様の仮定を用いて、階層鍵管理を行う。

【００３２】図１に示したように、各ホストコンピュータ（通信装置）Ａは自分の秘密鍵K_A^{-1}を保持しており、本籍のマネージャ（通信管理装置）Ｍの公開鍵K_Mを知っている。また、本籍のマネージャＭはそれに属する全ホストコンピュータの公開鍵のデータベースを持ち、また、自分の秘密鍵K_M^{-1}を保持している。同じ本籍を持つホストコンピュータが通信している限り、鍵管理は単純である。すなわち、マネージャＭは識別コード（または名前）ＩＤ（Identification）で指定されたホストコンピュータの公開鍵をマネージャＭの秘密鍵でサインして返す。各ホストコンピュータはマネージャＭの公開鍵を知っているので、公開鍵を取り出して相手ホストコンピュータを認証したり、秘密メッセージを送ったりできる。ホストコンピュータが異なるマネージャに属している場合には、いくつかのマネージャを順にたどっていって順番に公開鍵を得る必要がある。例えば、図１に示したホストコンピュータＡからホストコンピュータＤに秘密のメッセージを送るためにホストコンピュータＤの公開鍵を得たいとする。もしホストコンピュータＡが直接マネージャＴから公開鍵K_B を送ってもらったとすると、これは認証されないので、間違った公開鍵が送

られてきた可能性がある。

【００３３】そこで、下記手順、

（１）ホストコンピュータＡはマネージャＳから鍵$K_S$$^{-1}$（$K_R$）を送ってもらい、鍵$K_S$を使って読む、

（２）ホストコンピュータＡはマネージャＲから鍵$K_R$$^{-1}$（$K_T$）を送ってもらい、鍵$K_R$を使って読む、

（３）ホストコンピュータＡはマネージャＴから鍵$K_T$$^{-1}$（$K_B$）を送ってもらい、鍵$K_T$を使って読む、

を経れば、安全に公開鍵K_Bを得ることができる。

【００３４】ここで、各マネージャは「一つ向こうの」公開鍵を正しく教えると仮定している。つまり、

（ａ）ホストコンピュータＡは、「鍵K_Sで読むと、ホストコンピュータＡ以外のマネージャＳに隣接するホストコンピュータが言ったことになっている命題」を、そのホストコンピュータが本当に言ったと信じる。

（ｂ）マネージャＳは、「鍵K_Rで読むと、マネージャＳ以外のマネージャＲに隣接するホストコンピュータが言ったことになっている命題」を、そのホストコンピュータが本当に言ったと信じる。

（ｃ）マネージャＲは、「鍵K_Tで読むと、マネージャＲ以外のマネージャＴに隣接するホストコンピュータが言ったことになっている命題」を、そのホストコンピュータが本当に言ったと信じる。

ということである。これとマネージャＴのデータベースの情報である、「ホストコンピュータＤは、自分の公開鍵がK_Bであると言った」という命題を組み合わせることで、ホストコンピュータＡはホストコンピュータＤの公開鍵がK_Bであると信じるようになる。このやり方は上記文献１で形式的に述べられている。

【００３５】上記仮定が成り立たなかった場合、つまり、どこかのマネージャが嘘の公開鍵を返した場合、メッセージを相手が受け取ることができなくなってしまう。一方、マネージャは、メッセージの中継をごまかすことによってもメッセージを届かなくすることができる。つまり、マネージャが正しい公開鍵を返すかどうかは、マネージャを通るメッセージが正しく届けられるかどうかということと同程度に信用できる。換言すれば、従来の方法では、マネージャ（認証サーバ）は生成した共有鍵を用いてメッセージをこっそり盗聴することができ、これを検出するのは困難である。したがって、これは妥当な仮定である。

【００３６】次にマイグレーション（ホストコンピュータの移動）がある場合について述べる。ホストコンピュータが移動した場合、従来のやり方では、移動先でも本籍に認証してもらうか、または移動先でも認証してもらえるように本籍から手続きを行っておく必要があった。すると、ホストコンピュータの信用に関する仮定が増えてしまう。ところで、移動先で物理的接続がされるときには、そこのマネージャの公開鍵を直接入力することができる。また、同時にマネージャはホストコンピュータの公開鍵をデータベースに入れることができる。これを使うと、移動したホストコンピュータが移動先のホストコンピュータを認証する場合でも、その逆の場合でも、信用しなければならないホストコンピュータの数を減らすことができる。

【００３７】例えば図２でホストコンピュータＢが移動してマネージャＴと接続されたとする。そのときに、公開鍵K_TをホストコンピュータＢに入力し、また、マネージャＴのデータベースに鍵K_Bを登録すれば、ホストコンピュータＢは「鍵K_Tで読むと、ホストコンピュータＢ以外のマネージャＴに隣接するホストコンピュータが言ったことになっている命題」を、そのホストコンピュータが本当に言ったと信じるという仮定のもとで、そこのホストコンピュータ、例えばホストコンピュータＣの認証ができるようになる。したがって、上述した仮定（ａ）～（ｃ）は不要である。

【００３８】逆にいえば、移動先でホストコンピュータが特に公開鍵を直接入力することがなければ、本籍のマネージャから始めて今までと同じ仮定をおく必要があるし、相手がこちらを認証するには移動先のマネージャは本籍のマネージャに頼んで（やはり同様の仮定が必要）公開鍵を取り寄せなければならない。

【００３９】階層相対名前付け法におけるホストの移動手順に認証を付け加えた例を述べる。階層相対名前付け法では、本籍は常にそれに属するオブジェクトの現在位置を正しく知っている必要があるため、切断通知、新住所通知、確認通知には認証が加わった。この例では、移動するホストコンピュータをＡとする。

【００４０】（１）移動開始：ホストコンピュータＡは本籍のマネージャに切断通知を送る。これには、ホストコンピュータＡの秘密鍵でサインしたオブジェクト識別コードＯＩＤ（Object ID）とオブジェクトアドレスＯＡＤ（Object address）とを付け加えて認証する。このオブジェクトアドレスＯＡＤにはタイムスタンプが含まれているため、切断通知の再送は防止される。このメッセージが通過した名前空間のマネージャは、ホストＡとその子孫に対する局所識別コードＬＩＤ（Local ID）とオブジェクトアドレスＯＡＤ、オブジェクト識別コードＯＩＤと局所アドレスＬＡＤ（Local Address）、または、オブジェクト識別コードＯＩＤとオブジェクトアドレスＯＡＤの組を無効にする（これは認証されなくてもよい）。

【００４１】（２）移動：ホストコンピュータＡが移動する。

【００４２】（３）移動終了：新しい現住所のマネージャに局所アドレスＬＡＤを割り当ててもらう。マネージャはこのオブジェクト識別コードＯＩＤと局所アドレスＬＡＤの組を記憶する。このとき同時に、ホストコンピュータＡは現住所のマネージャの公開鍵を記憶し、また、ホストコンピュータＡの公開鍵を現住所のマネージ

ャのデータベースに登録することが望ましい。そして、ホストコンピュータAは、本籍のマネージャにオブジェクト識別コードＯＩＤ、新しいオブジェクトアドレスＯＡＤ、仮想的なオブジェクト識別コードＯＩＤである（０：）と、それらをホストコンピュータAの秘密鍵でサインしたものを通知する。本籍のマネージャはオブジェクトアドレスＯＡＤを更新し、確認通知を返す。確認通知には、新住所通知のオブジェクトアドレスＯＡＤについていたタイムスタンプと、仮想的なオブジェクト識別コードＯＩＤ０：の現在の値（逆ＯＩＤ）が、本籍の秘密鍵でサインされたものが含まれている。

【００４３】階層相対名前付け法の場合、識別コードＩＤが相対表現なので、本籍のマネージャと認証通信して相対位置を確認しないと識別コードＩＤが決められない。これはつまり、識別コードＩＤの扱いに関しては途中のホストコンピュータを信用しているということである。これは次の２つの点で問題がある。もし本籍のマネージャと通信できなかった場合どことも通信が始められないこと、および、依然として信用すべきホストコンピュータが減らないことである。このうち特に前者を解決する方法として、本籍のバックアップの働きをするホストコンピュータを用意する。後者の問題は、バックアップの識別コードＩＤの確定の問題があって解決し難いが、ホストコンピュータの移動前に移動先を決めておくことによってある程度解決できる。

【００４４】オブジェクトについてマイグレーションが起きる場合は、ホストコンピュータについてと同様に、オブジェクトも認証の対象にしなければならない。ただし、移動先のホストコンピュータは信用できなければならない。ホストコンピュータはオブジェクトの全データに（秘密鍵にも）アクセスできるからである。機密情報を交信するという点では、通常の通信もオブジェクトマイグレーションも相手のホストコンピュータを同程度に信用している必要がある。ただ、外から見た場合、移動先のホストコンピュータを認証するよりもオブジェクトそのものを認証できたほうが都合がよいこと、移動先からの新住所通知をオブジェクトの鍵で認証できれば便利なことから、オブジェクトにも秘密鍵と公開鍵の組を割り当てる。

【００４５】オブジェクトマイグレーションの手順は、ホストコンピュータの場合とほとんど同じである。オブジェクトが初めて本籍を離れるときに、マネージャがオブジェクトに秘密鍵を割り当て、公開鍵をデータベースに登録するという点が異なる。オブジェクトが本籍に帰るときには、公開鍵と対応する秘密鍵を破棄し、次回は別の鍵を使うこともできる。

【００４６】本発明の方法では、従来の方法に比べると、移動した先にあるホストコンピュータとの通信で用いる仮定が少なくなり、より信頼できる通信が可能である。さらに、移動先のデータベースの公開鍵を覚えたままにしておけば、ホストコンピュータが元の場所に戻ってからも、この信頼性は変わらない。

【００４７】以上に述べたように、本発明に基づく大規模分散システムに適した公開鍵認証のための鍵管理においては、鍵は階層管理され、鍵データベースへのアクセスの集中が防止でき、また、ホストコンピュータの移動を積極的に利用して、信用しなければならないホストコンピュータの数をできるだけ少なくした秘密・認証通信ができる。

【００４８】本発明の基本技術である公開鍵鍵暗号方式について述べる。平文とは、暗号化される前のデータをいい、暗号文とは、暗号化後のデータをいう。暗号変換式Ｅκは平文から暗号文への鍵Ｋによって決まる変換式であり、復号変換式Ｄκ'は、暗号文から平文への、鍵Ｋ'によって決まる変換式である。鍵から暗号化変換式、復号化変換式を決める手順は公開されているとする。各平文Ｍについて、

【数１】

$$D_{K'}(E_K(M))=M$$

である。

【００４９】ＤＥＳ（Data Encryption Standard, FIPS PUB 46, National Bureau of Standards, Washington, D.C. (Jan. 1977)）などの共有鍵暗号では、Ｋ＝Ｋ'であるが、1976年にDiffie とHellman によって紹介された下記文献、

文献６：W. Diffie and M. Hellman, "New Direction in Cryptography", IEEE Transactions on Informati on Theory Vol. IT-22(6) pp. 644-654, Nov1976

に記載された公開鍵暗号方式では、鍵Ｋと鍵Ｋ'とは異なり、鍵Ｋから鍵Ｋ'を求めることは非常に困難である。

【００５０】公開鍵暗号方式は、上述したように、暗号化に使う鍵（公開鍵）と解読に使う鍵（秘密鍵）を異なるものとし公開鍵から秘密鍵を推測し難くした暗号処理方式であり、この公開鍵暗号を用いて秘密通信を行うシステムでは、各ユーザ（ホストコンピュータ）Ａは２つの鍵K_Aと鍵K_A'を持っている。鍵K_Aは公開鍵と呼ばれ、公開鍵データベースに登録されている。公開鍵データベースは、ユーザ名Ａを指定するとその公開鍵K_Aを答える機能を持つ。鍵K_A'は秘密鍵と呼ばれ、ユーザＡのみが知っている。ユーザＡが平文Ｍを秘密裡にユーザＢに送ろうとするときには、ユーザＡは公開鍵データベースから鍵K_Bを得、ユーザＢに暗号文

【数２】

$$C=E_{K_B}(M)$$

を送る。ユーザＢは、それを受信して平文

【数３】

$$M = D_{K_B'}(C)$$

を得る。秘密鍵K_B'はユーザBのみが知っており、しかも、鍵K_Bから鍵K_B'を計算するのは非常に困難であるため、ユーザB以外のユーザが平文Mを得るのを防ぐことができる。

【００５１】公開鍵暗号を用いて認証（本人確認）を行うシステムでは、暗号化変換と復号化変換に対する仮定として、平文を復号化変換できること、暗号文を暗号化変換できること、各平文Mについて

【数４】

$$M = E_K(D_{K'}(M))$$

であることを要請する。上と同様に各ユーザが二つの鍵を持ち、公開鍵データベースを用意するならば、ユーザAが平文MをユーザBに、確かにユーザAから送られたことがわかるように送るときには、ユーザAはユーザBに

【数５】

$$C = D_{K_A'}(M)$$

を送る。ユーザへはそれを受信して後、あるいはあらかじめ公開鍵データベースから鍵K_Aを得、それを用いて平文

【数６】

$$M = E_{K_A}(C)$$

を得る。秘密鍵K_A'はユーザAのみが知っており、しかも、鍵K_Aから鍵K_A'を計算するのは非常に困難であるため、意味のある平文に復元できるようなコードを作れるのはユーザAだけである。つまり、ユーザA以外のユーザがユーザAのふりをして通信することはできない。

【００５２】公開鍵暗号の具体例としては、

文献７：R. L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM Vol 21(2) pp. 120-126, Feb. 1978

に記載されたＲＡＳ方式が知られている。

【００５３】非常に多くのコンピュータを含むネットワークシステムの場合、ユーザ数も非常に多く、公開鍵データベースをシステム内に１個だけ用意しておいた場合、データベースは巨大になり、また、データベースへのアクセス頻度が非常に高くなる。これを避けるために公開鍵データベースは分散管理するのが普通である。例えば、上述した文献１（Butler Lampson, Martin Abadi, Michael Burrows, and Edward Wobber, "Authentication in Distributed Systems: Theory and Practice", Proceedings of the 13th ACM Symposium on Operating System Principles, October 1991）に記載されたLampson らのシステムの一例を次に示す。

【００５４】図３で、記号C_0, C_1, C_2, …C_{11}, C_{12}, …C_{21}, C_{22}, …は通信を行うコンピュータ（計算機）を示しており、鍵K，K'から暗号化変換式E_Kと復号化変換式$D_{K'}$とを求め、それを実行することができる。ここでは、ネットワークシステムに接続されているコンピュータが３段の木構造に管理されているとする。つまり、記号C_1はコンピュータC_1, C_{11}, C_{12}, …を代表するコンピュータを示しており、コンピュータC_{11}, C_{12}, …の公開鍵K_{11}, K_{12}, …を記憶している公開鍵データベースDB_1を持っている。同様に、コンピュータC_2, はコンピュータC_2, C_{21} C_{22}, …を代表するコンピュータであり、コンピュータC_{21}, C_{22}, …の公開鍵K_{21}, K_{22}, …を記憶している公開鍵データベースDB_2を持っている。また、コンピュータC_0はコンピュータC_0, C_1, C_2, …を代表するコンピュータであり、コンピュータC_1, C_2, …の公開鍵K_1, K_2, …を記憶している公開鍵データベースDB_0を持っている。各コンピュータは一人のユーザが使用しているとする。つまり、各コンピュータC_nは公開鍵K_nと秘密鍵K_n'を記憶している。公開鍵を公開鍵データベースに登録する作業は、人手で本人確認をして行う。

【００５５】このシステムでは、公開鍵データベースから公開鍵を得るときにもネットワークを使用する。従って、公開鍵データベースを持つコンピュータ以外から誤の公開鍵を与えられることを防ぐために、公開鍵データベースも認証の対象にする。例えば、コンピュータC_{11}には、自分の公開鍵と秘密鍵のほかに公開鍵データベースDB_1を持つコンピュータC_1の公開鍵K_1が予め人手で入力されている。コンピュータC_{12}の公開鍵を得たい場合、「コンピュータC_{12}の公開鍵はK_{12}である。」という通信文をM_{12}とすれば、コンピュータC_1からコンピュータC_{11}に復号文$D_{K1}'(M_{12})$を送ってもらえばよい。コンピュータC_{11}は公開鍵K_1を用いて通信文$M_{12}=E_{K1}(D_{K1}'(M_{12}))$を得、確かにコンピュータ$C_1$から送られたものと確認することができる。ここでは、コンピュータC_1は十分に信頼でき、コンピュータC_{11}, C_{12}, …の公開鍵K_{11}, K_{12}, …を正しく教えると仮定している。

【００５６】代表となるコンピュータが異なるようなコンピュータの公開鍵を得ようとする場合は、いくつかの公開鍵データベースを順に検索していく必要がある。例えば、コンピュータC_{11}がコンピュータC_{21}の公開鍵を得たい場合、「コンピュータC_0の公開鍵はK_0である。」という通信文をM_0とすれば、まずコンピュータ

C_1 に頼んでD_{K1}'（M_0）を送ってもらう。コンピュータC_{11}は公開鍵K_1を用いて$M_0 = E_{K1}(D_{K1}'(M_0))$を得る。次に、「コンピュータ$C_2$の公開鍵は$K_2$である。」という通信文を$M_2$とするとき、コンピュータ$C_0$に頼んで$D_{K0}(M_2)$を送ってもらう。コンピュータ$C_{11}$は先ほど得た公開鍵$K_0$を用いて$M_2 = E_{K0}(D_{K0}(M_2))$を得る。最後に、「コンピュータ$C_{21}$の公開鍵は$K_{21}$である。」という通信文を$M_{21}$とするとき、コンピュータ$C_2$に頼んで$D_{K2}'(M_{21})$を送ってもらう。コンピュータ$C_{11}$は先ほど得た公開鍵$K_2$を用いて$M_{21} = E_{K2}(D_{K2}'(M_{21}))$を得る。こうしてコンピュータ$C_{21}$の公開鍵$K_{21}$を得ることができる。ここでは、コンピュータ$C_1, C_0, C_2$,は十分に信頼でき、その一つ向こうのコンピュータの公開鍵を正しく教えると仮定している。

【００５７】信頼するべき経路を辿るコンピュータの数を減らしたい場合には、公開鍵をあらかじめ知らせてあるコンピュータを、木構造で管理されている関係より増やせばよい。例えば、コンピュータC_2の公開鍵K_2をあらかじめ人手でコンピュータC_1に入力しておけば、コンピュータC_0を信頼するという仮定は必要なくなる。ただし、コンピュータC_1がコンピュータC_2の公開鍵を知っているという情報は、何らかの形でコンピュータC_{11}が知る必要がある。

【００５８】文献１に示したLampson らのシステムの問題点は、上述したように移動可能なコンピュータについて考慮されていないことである。例えば、コンピュータC_{11}が移動可能な計算機であって、コンピュータC_{11}のユーザがコンピュータC_{11}をコンピュータC_{21}の近くに持って行ったとする。このときでもコンピュータC_{11}からコンピュータC_{21}に秘密の通信文を送ろうとするときには、コンピュータC_1, C_0, C_2と順に通信してコンピュータC_{21}の公開鍵K_{21}を得なければならないし、コンピュータC_1, C_0, C_2は信頼できると仮定しなければならない。逆に、コンピュータC_{21}からコンピュータC_{11}に秘密の通信文を送ろうとするときにも、コンピュータC_2, C_0, C_1と順に通信してコンピュータC_{11}の公開鍵K_{11}を得なければならないし、コンピュータC_2, C_0, C_1は信頼できると仮定しなければならない。

【００５９】本発明では、上の問題を解決するために、各コンピュータはそれまでに公開鍵を知ったコンピュータの名称とその公開鍵、そしてそれを知るために信頼したコンピュータの名称の対応表（テーブル）を持っているとする。対応表は公開鍵データベースから公開鍵を教えてもらったときのほか、コンピュータのユーザがコンピュータの名称とその公開鍵を入力したとき（このときには信頼したコンピュータの名称は空である）にも更新される。特に、移動可能なコンピュータの場合には、他のコンピュータの近くに移動したときに、そのコンピュータのユーザから直接公開鍵を教えてもらえることが期待できるので、信頼したコンピュータの数の少ない公開鍵を得ることができる。

【００６０】例えば、図４でコンピュータC_{11}のユーザがコンピュータC_{11}をコンピュータC_{21}の近くに持って行ったときに、コンピュータC_{21}のユーザからその公開鍵K_{21}を教えてもらったとする。すると、他のコンピュータを信頼することなく、コンピュータC_{11}からコンピュータC_{21}に秘密の通信文を送ることができる。この状況は、コンピュータC_{11}を元の場所に持って帰ったり、さらに他の場所に移動したりしてからでも変わらない。また逆に、コンピュータC_{11}のユーザがコンピュータC_{21}のユーザに公開鍵K_{11}を教えれば、他のコンピュータを信頼することなく、C_{21}からC_{11}に秘密の通信文を送ることができるし、コンピュータC_{11}をまた移動してからでも同様である。

【００６１】他の例として、コンピュータC_{11}をコンピュータC_2の近くに持って行き、公開鍵K_2を入力したとする。すると、コンピュータC_{22}に秘密の通信文を送りたい場合、すでに対応表に、「コンピュータC_2の公開鍵はK_2で、信頼したコンピュータはない」という情報があるため、コンピュータC_2と通信して公開鍵K_{22}を教えてもらうだけでよい。この場合、信頼したコンピュータはコンピュータC_2のみである。このようにして、公開鍵を直接入力したコンピュータを中心としたいくつかのコンピュータについては、その公開鍵を得るために信頼したコンピュータの数を、Lampson らのシステムに比べて少なくすることができる。

【００６２】公開鍵を得たいコンピュータの名前と、すでに対応表に記録されているコンピュータの名前から、どのコンピュータと順に通信していくかを決定するために、コンピュータの名前付けはコンピュータの管理の木構造、つまり、鍵管理の木構造を反映したものでなければならない。これは、Lampson らのシステムでも同様である。

【００６３】図４で、コンピュータC_0, C_1, C_2,…は据え置き型コンピュータであり、$C_{11}, C_{12}, \cdots C_{21}, C_{22}$, …は据え置き型、あるいは移動可能なコンピュータである。各コンピュータは、鍵K，K’から暗号化変換式Ekと復号化変換式Dk’を求め、それを実行することができる。これらのコンピュータはコンピュータネットワークシステムに含まれており、各コンピュータは、システム内の他のコンピュータに通信文を送ることができる。コンピュータが移動しても通信文は相手のコンピュータに到着するものとする。ネットワークシステムの各通信路では、通信文の送信者と受信者以外の者に通信内容を読み取られたり、通信内容を改変されたりする可能性がある。

【００６４】コンピュータネットワークシステムに接続されているコンピュータは、鍵管理の３段の木構造に管理されているとする。つまり、コンピュータC_1はコン

ピュータC_1, C_{11}, C_{12}, …を代表するコンピュータであり、コンピュータC_{11}, C_{12}, …の公開鍵K_{11}, K_{12}, …を記憶している公開鍵データベースＤＢ$_1$を持っている。同様に、コンピュータC_2はコンピュータC_1, C_{21}, C_{22}, …を代表するコンピュータであり、コンピュータC_{21}, C_{22}, …の公開鍵K_{21}, K_{22}, を記憶している公開鍵データベースＤＢ$_2$を持っている。また、コンピュータC_0はコンピュータC_1, C_2 …を代表するコンピュータであり、コンピュータC_1, C_2…の公開鍵K_1, K_2, …を記憶している公開鍵データベースＤＢ$_0$を持っている。各コンピュータは一人のユーザが使用しているとする。つまり、各コンピュータC_nは公開鍵K_nと秘密鍵K_n'を記憶している。公開鍵を公開鍵データベースに登録する作業は、人手で本人確認をして行う。

【００６５】各データベースが管理しているコンピュータと鍵の数はたとえば、１０００～１００００程度とする。公開鍵暗号としてＲＳＡ方式を用いた場合、公開鍵は２００桁の１０進数２個（１，３キロビット）程度の記憶領域を必要とするため、データベースの大きさは１．３～１３メガビット程度となる。

【００６６】各コンピュータC_0, C_1, C_2, （一般的にはC_nとして表す）は、表（テーブル）T_0, T_1, T_2, （一般的にはT_n として表す）を持つ。対応表の各行には、コンピュータの名前（名称）、公開鍵、使われた時刻、信頼したコンピュータの名称（集合）が記録されている。対応表の行数は１００～１０００程度とし、それを越える行を記憶させようとした場合には、信頼したコンピュータの名称の数が多い順、数が同じものが複数あるときには時刻の古い順に消去する。つまり、新しいものを残しておく。コンピュータの名称の大きさなどを無視すれば、対応表の大きさは１３０キロビット～１．３メガビット程度となる。

【００６７】コンピュータC_nのユーザがコンピュータC_pの公開鍵K_pを入力したときの表更新手順を図５に示す。まず、対応表T_nにコンピュータC_p についての行があるとき（ステップ１）は、コンピュータC_p についての行を作成する（ステップ２）。ないときには、その行の公開鍵とK_pを比べる（ステップ３）。異なる場合には、信頼したコンピュータのうちのどれかが嘘をついたことがわかるため、その旨エラー表示した後（ステップ４）、信頼したコンピュータの名称にコンピュータC_pを含む、対応表T_nのすべての行を削除する（ステップ５）。一致したなら、その行の信頼したコンピュータの名称をC, C' C'' …とするとき（ステップ６）、対応表T_nの各行のうち、信頼したコンピュータの名称にC_p, C, C', C'' …が含まれるなら、コンピュータC, C', C'' …を除く（ステップ７）。最後にコンピュータC_p についての行の公開鍵にK_pを、使われた時刻に現在時刻を、信頼した計算機名に空集合を設定する（ステップ８）。

【００６８】コンピュータC_n のユーザがC_p の公開鍵を得ようとしたときの表更新手順を図６に示す。まず、コンピュータC_p がコンピュータC_n の代表となるコンピュータであった場合は、すでに公開鍵を知っているので終了する（ステップ１１）。対応表T_n にコンピュータC_p についての行があるとき（ステップ１２）はそれを使えばよいので終了する。ないときには、対応表T_nの各行と代表となるコンピュータについて、コンピュータC_pから木構造をたどったときの段数と信頼したコンピュータの数（代表となるコンピュータの場合は０）の和を計算し（ステップ１３）、和の最も少ないコンピュータをC_q とする（ステップ１４）。コンピュータの数が複数あるときは段数の少ないものから任意に一つを選ぶ。コンピュータC_q に木構造で隣接し、コンピュータC_p に一段近いコンピュータ（一意に決まる）をコンピュータC_r とする（ステップ１５）。対応表T_n のコンピュータC_q についての行の使われた時刻を現在時刻とする（ステップ１６）。コンピュータC_q と通信してコンピュータC_r の公開鍵K_r を得る（ステップ１７）。コンピュータC_q についての行の信頼したコンピュータの名称をC, C', C'' …とする（ステップ１８）。対応表T_n にコンピュータC_r についての行を作成し（ステップ１９）、公開鍵にK_r を、使われた時刻に現在時刻を、信頼したコンピュータの名称にコンピュータC, C', C'' …を設定する（ステップ２０）。そしてステップ１２に戻る。

【００６９】以上のように処理することにより、ホストコンピュータの移動、あるいは、オブジェクトの移動があっても、大規模分散コンピュータ通信ネットワークシステムにおいて、信頼するコンピュータの数を減らして信頼性の高い通信を行うことができる。

【００７０】

【発明の効果】本発明によれば、ユーザが望むならば直接公開鍵をやりとりすることで、大規模分散システムでも、小規模なシステムと同様の信頼性を得ることができる。また、それほど信頼性を要求しない場合は、間接的に公開鍵を得ることもでき、ユーザの要求レベルに合わせた信頼性で秘密通信・認証が可能になる。

【００７１】また本発明に基づく大規模分散システムに適した公開鍵認証のための鍵管理においては、鍵は階層管理され、鍵データベースへのアクセスの集中が防止でき、また、ホストコンピュータの移動を頻繁に利用して、信用しなければならないホストコンピュータの数をできるだけ少なくした秘密・認証通信ができる。

【００７２】さらに本発明の方式は、従来の方式に比べると、移動した先にあるホストコンピュータとの通信で用いる仮定が少なくなり、より信頼できる通信が可能である。さらに、移動先のデータベースの公開鍵を覚えたままにしておけば、ホストコンピュータが元の場所に戻ってからも、この信頼性は変わらない。

【図面の簡単な説明】

【図１】従来の公開鍵暗号処理システムの構成図である。

【図２】本発明の公開鍵暗号処理システムの構成図である。

【図３】従来の公開鍵暗号処理システムの構成図である。

【図４】本発明の公開鍵暗号処理システムの構成図である。

【図５】図４に示した公開鍵暗号処理システムにおける公開鍵を直接入力する場合の処理方法を示すフローチャートである。

【図６】図４に示した公開鍵暗号処理システムにおける公開鍵を入手するときの更新手順を示すフローチャートである。

【符号の説明】

C_0，C_1，C_2，C_n・・通信を行うコンピュータ

T_0，T_1，T_2，T_n・・通信を行うコンピュータ内の対応表

DB_0，DB_1，DB_2，DB_n・・データベース

K_0，K_1，K_2，K_n・・鍵